Aalto University
School of Economics

Value Creation and Value Capture in Software Product Business: Analyzing Product Development, B2B Sales and Software Process Methodologies

Marketing

Master's thesis

Antti Juneja

2011

Department of Marketing
Aalto University
School of Economics

Abstract

Research target

This study is about value creation and value capture in B2B software product business in new product development context. The goal of the study is to understand how academic and managerial new product development, B2B sales and software process methodologies fit with the value creation and value capture theory.

Research methods

Research method is desk research. This study breaks new product development, B2B sales and software process methodologies in pieces and analyzes what are the principles for value creation, what actors are involved and how learning and risk are treated.

Results

The results of this study are two-folded. Firstly the value creation and value capture and dynamic capability are relatively weak concepts to explain practical innovation work in companies. Secondly, methodologies analyzed are partially conflicting and none of them cover the full process how innovation should be understood and managed. Main issues in are how to reach dominant design and mainstream market in order to reach profits, how to arrange sales channel and how to apply software process properly.

Keywords

Product development, innovation, B2B sales, marketing, software process, agile, waterfall, dynamic capability, value creation, value capture, Stage-Gate, mainstream market.

TABLE OF CONTENTS

1. Introduction

Recently studies have begun to suggest that sustainable competitive advantage is rare and declining in duration and temporary advantage explains competition better nowadays. This is called hypercompetition (D'Aveni 1994, D'Aveni, Dagnino & Smith 2010). The software industry is often cited as the epitome of high-technology industries where hypercompetition is experienced (Lee, Venkatraman, Tanriverdi & Iyer 2010). Software industry is characterized by high-velocity innovation (Brown & Eisenhardt 1997), technological change (Schmalensee, 2000), and turbulence in revenues, market shares, and profits of firms (D'Aveni, Dagnino & Smith 2010, Schmalensee 2000, Shapiro & Varian 1999).

As competitive cycles have shortened, the need to rapidly develop new advantages (products) has increased. It has become more important for companies to focus on generating their next advantages even before their current advantages erode. The traditional goal of strategists has been to find a grand and long-term strategy that sustains itself for years or even decades. Instead of long-range plans and enduring competitive advantages, a succession of small, often easily duplicated strategic attacks are more typically used in rapidly changing hypercompetitive environments (D'Aveni 1994, D'Aveni 1995, see also *Figure 1*). By stringing together a series of these short-term advantages, the firm can effectively create a long-term sustainable advantage in the marketplace but it requires efficient management of innovation processes.

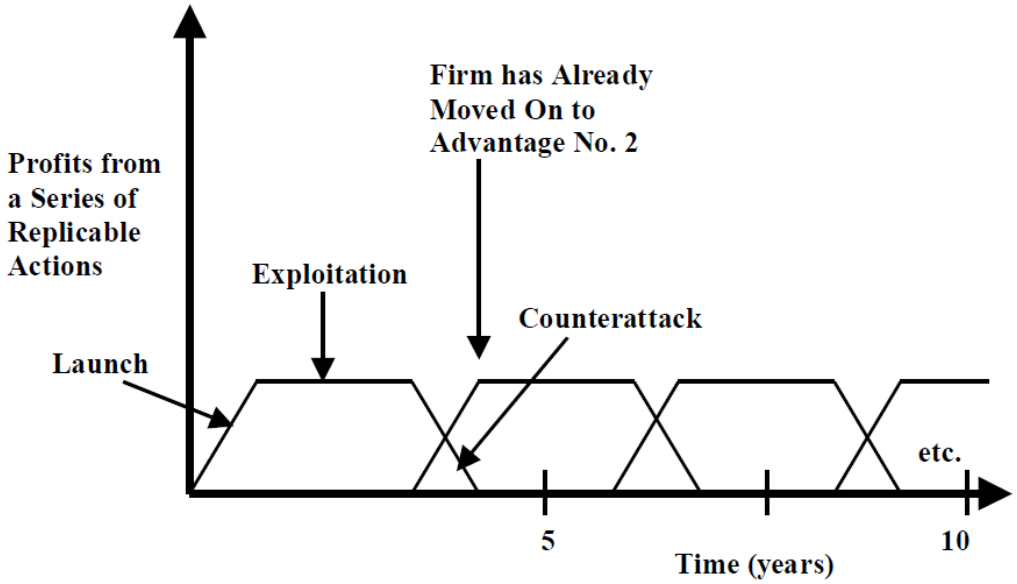

Figure 1 - Hypercompetition - series of short term advantages (D'Aveni 1994 pp. 10)

Innovation is the process how companies can survive and prosper over long time period due to constantly changing market demand and competitive situation (Trott 2005 pp.5, Cooper 2001 ch. 1). Innovation is also very important on national level because innovation is seen as source of welfare and governments typically have many programs to help companies with innovation activity (Arnold & Bell 2001, Arnold 2004).

Companies are forced to look new sources of profit in order share to the profit for growing demands by owners, employees and suppliers. However, companies perform very differently with their innovation results (Arthur D Little 2005, Cooper & Edgett 2008).

Classical study by Joseph Schumpeter (Schumpeter 1950) explains that firms should seek Creative Destruction in their innovation activity (for definition see *Table 1* below). Main idea of Creative Destruction is that companies seek to gain monopoly profits by creating innovations that destroy current business models and create new ones. Creative Destruction creates economic

discontinuities, and in doing so, creates also an entrepreneurial environment for the introduction of innovation and makes monopoly profits possible.

Table 1 – Main elements of Creative Destruction (adapted from Schumpeter 1950)

#	Description
1.	Competition is a self-destructive mechanism that normalizes the profit level when the innovation effects have been utilized.
2.	From company's perspective this means that if competitors have created similar offerings and capabilities, company's competitive advantage has been lost.
3.	Creative Destruction is continuously on-going process and gives reasoning for companies acting as entrepreneurs. Entrepreneurs create innovations for overcoming the competition and thereby create irregular economic growth.
4.	Companies should target superior price-performance for customers in order to trigger Creative Destruction (Spencer & Kirchhoff 2006)

From company owners' perspective primary measure of innovation success is creation of profit. Creating profit is measured by return on invested capital (ROIC). Financially, target of company's innovation process is to create level of ROIC in order to exceed alternative investment options. Otherwise company is destroying the capital.

This study examines the innovation process, meaning value creation and value capture, in business-to-business (B2B) software product business. The need to understand innovation process and managing it efficiently is crucial for companies to survive and prosper in today's competitive landscape.

2. Research Problem and Objectives

Value creation and value capture have already existing research tradition (see Bowman & Ambrosini 2000, Bowman & Ambrosini 2007, Lepak, Smith & Taylor 2007 and O'Cass & Ngo 2009). However, based on my 11 years of practical experience in B2B software product business in various roles from software engineer to product business owner and as sales director, I see that current value creation and value capture literature fails to explain the value creation of software product company. I believe that this is partly because of current literature is generic and doesn't really focus on software product business but also that current literature doesn't take into account other viewpoints and managerial practices that are already well-established in the industry.

Currently value creation and value capture literature takes very much resource-based view (RBV) of the firm and stating that competitive advantage comes from dynamic capabilities and embedding them with valuable, rare, in-imitable and non-substitutable (VRIN) resources (Teece, Pisano & Shuen 1997, Bowman & Ambrosini 2009). Without taking into account other viewpoints than RBV and excluding best-of-breed management practices of a software product company value creation and value capture model can only, at best, be high-level conceptual and superficial model without really explaining the processes.

This study aims to sophisticate current academic understanding of value creation and value capture by analyzing academic and managerial innovations models, B2B sales process and modern software development processes. As each of the models and processes have their own roots and principles they are partially complementing but at the same time conflicting. Idea is to analyze these principles and conflicts and map them with value creation and value capture model.

2.1 Understanding of Value

To this end understanding what value is and how value is created has attracted significant attention over the past decade (O'Cass & Ngo 2009, Anderson, Narus & van Rossum 2006, Bowman and Ambrosini 2000 & 2009, Lepak, Smith & Taylor 2007, Möller 2006, Payne & Holt 2001, Sirmon, Hitt and Ireland 2007).

Resource-based view assumes that the core sources of value creation are valuable, rare, difficult to imitate and non-substitutable (VRIN) resources that are embedded within firm activities (Bowman & Ambrosini 2009). Current academic literature of value creation and value capture also makes difference between creation of value and the capturing of value (Bowman & Ambrosini 2000). Please, see *Table 2* below for definition.

Table 2 - Definition of value creation and value capture (adapted from Bowman & Ambrosini 2000)

Value Creation	• Supplier develops products that supplier believes have use value for customer. • Supplier uses its marketing and sales efforts to explain to customer what the use value is • Customer assess subjectively what the use value would be by using own criteria in making purchase decisions
Value Capture	• Value capture happens when sales realizes. The amount of captured value is defined as exchange value i.e. price paid for the use value (product) • Amount of exchange value is determined by the bargaining power of the parties

Figure 2 below explains the value creation (use value) and value capture (exchange value) process. Figure covers creation of use value and capturing exchange value in various levels in supply chain.

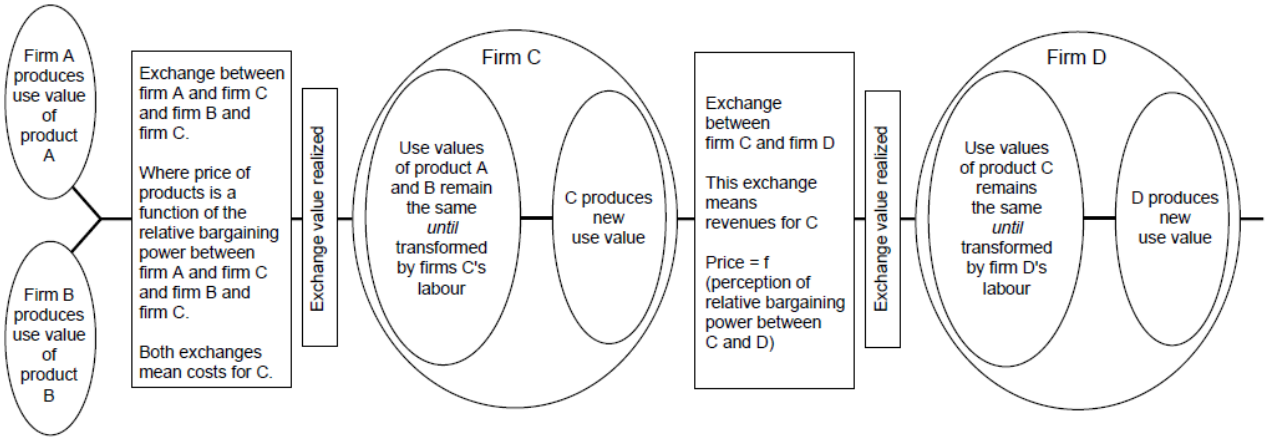

Figure 2 - Value Creation and Value Capture (Bowman & Ambrosini 2000, pp.12)

Value creation doesn't only happen in customer interface in a form of commercial transaction. Value can be created in R&D, procurement processes and so forth. Bowman and Ambrosini (2007) evolved the original value creation model to cover five different value creating activities within firm (see *Table 3* below). This thesis focuses only to first two value creating activities; R&D and Sales and Marketing.

Table 3 - Firm's value creating activities (adapted from Bowman & Ambrosini 2007)

#	Value creating activity
1.	R&D - creation of products/services
2.	Sales and Marketing - realising revenues from customers
3.	Capital market activities – financing the value creation. Financing strategy and execution to finance firms operations and strategic initiatives
4.	Procurement and Supply Chain Management - minimising cost flows to supplier
5.	Maintenance of firm – infrastructure and stakeholder activities. E.g. accounts preparation, legal work, tax management, etc. (necessary activities to conduct business, but they do not contribute to profit streams)

Scholars indicate that value creation is a dynamic and multi-stage process involving different users of value (Bowman & Ambrosini 2000, Lepak, Smith & Taylor 2007). Value creation and its management are important to both the firm and the customer, and need to account for different points in time in the process (Bowman and Ambrosini, 2000). Thus, value is created by dynamic capabilities.

Dynamic capabilities is broad topic and tightly linked to value creation and value capture research. Dynamic capability concept was originally introduced by Teece, Pisano & Shuen (1997) and since then the research community has been active and several definitions for dynamic capabilities have been created (see *Table 4* below).

Table 4 - Different definitions for dynamic capabilities (adapted from Bowman & Ambrosini 2009)

#	Definition
1.	Dynamic capabilities are 'The firm's processes that use resources – specifically the processes to integrate, reconfigure, gain and release resources – to match or even create market change. Dynamic capabilities thus are the organizational and strategic routines by which firms achieve new resources configurations as markets emerge, collide, split, evolve and die' (Eisenhardt and Martin 2000).
2.	A dynamic capability is a learned and stable pattern of collective activity through which the organization systematically generates and modifies its operating routines in pursuit of improved effectiveness' (Zollo & Winter 2002)
3.	Dynamic capabilities 'are those that operate to extend, modify or create ordinary capabilities (Winter 2003).
4.	Dynamic capabilities are 'the abilities to reconfigure a firm's resources and routines in the manner envisioned and deemed appropriate by its principal decision-maker' (Zahra et, Sapienza & Davidsson 2006).
5.	Dynamic capabilities are 'firm's behavioural orientation constantly to integrate, reconfigure, renew and recreate its resources and capabilities and, most importantly, upgrade and reconstruct its core capabilities in response to the changing environment to attain and sustain competitive advantage' (Wang & Ahmed 2007)
6.	Dynamic capability is 'the capacity of an organization to purposefully create, extend or modify its resource base' (Helfat et al. 2007)
7	Three dynamic capabilities are necessary in order to meet the business challenges, succeed and outpace the competition. (Teece, Pisano & Shuen 1997, and Eisenhardt & Martin 2000) 1. Organizations and their employees need the capability to learn quickly and to build strategic assets 2. New strategic assets such as capability, technology and customer feedback have to be integrated within the company 3. Existing strategic assets have to be transformed or reconfigured.

As seen in *Table 4* above, dynamic capability is clearly a process. As with all processes inside a firm dynamic capability needs to be managed properly in order to succeed. Recently, Bowman and Ambrosini (2009) have evolved dynamic capability model with included value creation process (see *Figure 3* below). Basically, model combines RBV view of value creation by dynamic

capabilities (company internal view) and Industrial Organization (IO) view of competition (external view, see e.g. Porter 1979, 1987) including path dependencies. Model is aggregate of existing models to explain competition and innovation.

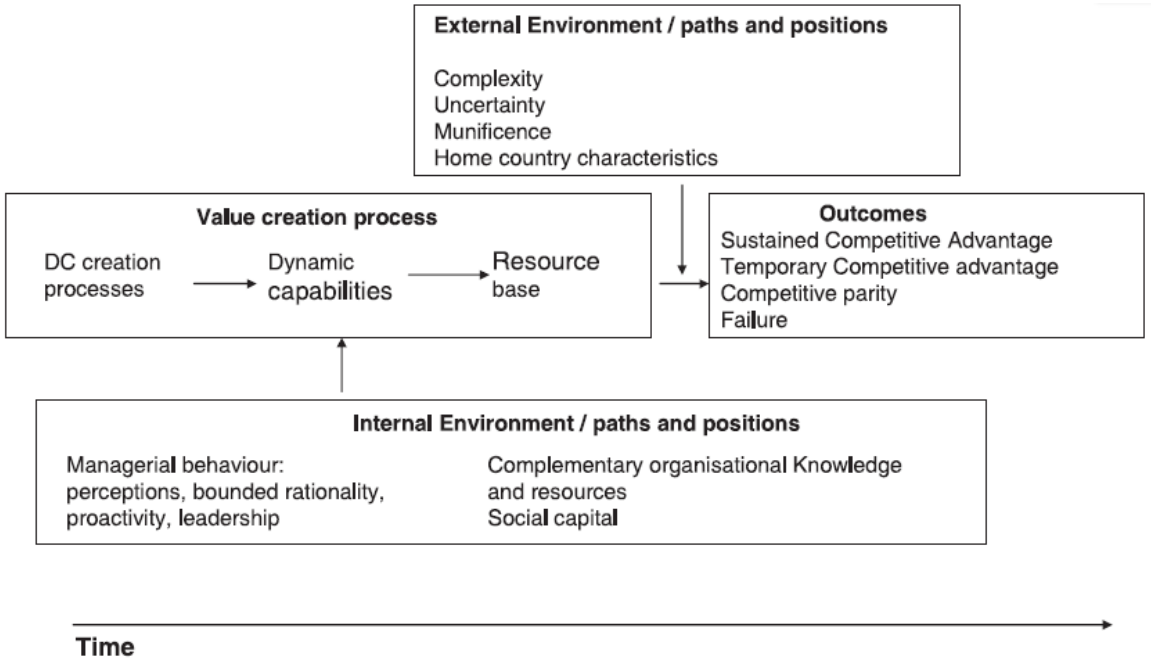

Figure 3 - Dynamic Capabilities integrated with value creation, RBV and IO (Bowman & Ambrosini 2009 pp.43)

Research in software engineering have also suggested that software process itself is dynamic capability (Mäkelä, Oza & Kontio 2009). This is due the software process is an industrial activity that requires high technological expertise and process and project management capabilities at many levels. A variety of qualities are required from the organization and the individuals engaged in software development and maintenance in order for them to succeed.

In the area of product development, there have only been few papers that their title includes “dynamic capabilities”. Deeds, DeCarolis & Coombs (2000) studied resource-based capabilities related hypotheses where geographical location, focused R&D efforts and CEO experience where mapped to the

success of high-tech ventures. Marsh and Stock (2003) developed a conceptual model of a process that collects, interprets and internalizes marketing and technological capabilities from previous product development projects and uses the resulting knowledge for the benefit of future product development projects.

This study is unique in a way as it maps both academic models with best-of-breed managerial processes concerning innovation in today's B2B software product companies.

2.2 Issues in Current View of Value Creation

Main issue with value creation and value capture literature is that it doesn't identify the practical processes that companies do in their day-to-day operations. Same applies with dynamic capabilities model, which is many times discussed with value creation and value capture theory. Synthesis of quite many definitions of dynamic capability is that it is purposefully managed process to create value by integrating, reconfiguring and releasing resources (see *Table 4* in previous section). However, as long as practical processes in companies are not identified, value creation and value capture including dynamic capability theory is, at best, philosophical framework and not giving any practical means for managers to guide decision making and operations better.

On the other hand, there is long history in managerial innovation, B2B sales and software development process literature that deal with practical processes and issues in day-to-day life in companies. Each of these models are having different history. Innovation and new product development process models have been formed up in long history of companies engaging in new product development activities with some help from academic research. Same applies with software development processes where practical know-how of doing world class software development has been mostly within companies. B2B sales processes have almost been purely developed by professional sales people by writing down their own know-how and experiences.

The current definitions of dynamic capability tend to be tautological i.e. acting dynamically and having dynamic capabilities to create value leads to (sustainable) competitive advantage. Bowman & Ambrosini (2009) summarized the status of value creation and value capture and dynamic capabilities research:

> *"To avoid the problems of tautology mentioned earlier, for dynamic capabilities to be a useful construct it must be feasible to identify discrete processes inside the firm that can be unambiguously causally linked to resource creation → thus value creation."*

> *"So by looking at the detail of how dynamic capabilities are deployed, we should be able to understand better the dynamic capabilities in practice and whether and how they might differ across firms, which could form the basis for developing managerial prescriptions."*

2.3 Research Question and Methods

This study focuses to R&D activities and Sales and Marketing while excluding the other activities in value creation (see *Table 3* in *section 2.1*). There are many types of innovations but this study concentrates on companies whose primary goal is to innovate through new software technology i.e. major component of new growth is based on software technology and not, for example, on consultancy services.

In addition, for the purpose of this study we define innovation as following (Trott 2005 pp.15):

- Innovation = new idea + technical invention + commercial exploitation

Above definition means that innovation includes the technical development as well as commercialization. Innovation depends on inventions but innovations need to be harnessed to commercial activities before they can contribute to commercial growth of companies. Innovation is the management of all the

activities involved in the process of idea generation, technology development and marketing of a new product.

Research questions of this thesis are formalized as following:

- How academic models and best-of-breed management practices fit with the value creation and value capture theory in the context of new product development in B2B software product business?
- Sub-questions are
    - How these models address the value creation and value capture, and
    - How these models are mutually overlapping and conflicting?

This study is desk research and is based on academic and managerial literature review. Method is to analyze different journal articles and management practices against research questions. Study doesn't include any interviews or case examples. *Table 5* below lists the main academic or managerial principles used for analysis in this study. Selection is made by the author based on expertise and judgment.

Table 5 - Sources used for analysis

Area	Main principles and source
Innovation and new product development	• Definition of innovation (academic) • Creative Destruction and targets for innovation (academic) • Innovation performance differences (academic and managerial) • Historical evolution of innovation models (academic) • Dominant design theory (academic) • Stage-Gate model (academic and managerial)
Business to business sales	• Sales message (managerial) • Crossing the Chasm (managerial) • Reasons for buying (managerial)
Software Development	• Software evolution and discontinuity (academic) • Waterfall process (academic) • Agile process (academic and managerial)

3. About Innovation Models

There is lot of written about innovation. Popular management literature focuses to this topic and gives various guidance to companies. For example, Kim & Mauborgne (2005) tell that companies should not compete head to head with similar value propositions as competitors. Companies can only succeed and earn more profit by creating new markets by new and differentiating value propositions. Collins (2001) tells that companies' innovation is successful only if it creates real profit to company, company can be best in the world, and employees of the company love what they are doing.

Innovation models have evolved during the history. The following sections discuss the history and evolution of innovation models in academia. Stage-Gate model, which is widely used innovation process model used in enterprises is briefly introduced here as it is one of key models used in this thesis. Also, analysis summary is presented of comparison of historical innovation models with Stage-Gate.

3.1 Historical Evolution of Innovation Models

3.1.1 Traditional Linear Models

After Second World War until the 1980s innovation was seen as linear more either technology push or market pull (Trott 2005 pp. 23). Linear models have been seen as sequence a separate stages and activities (see *Figure 4* below).

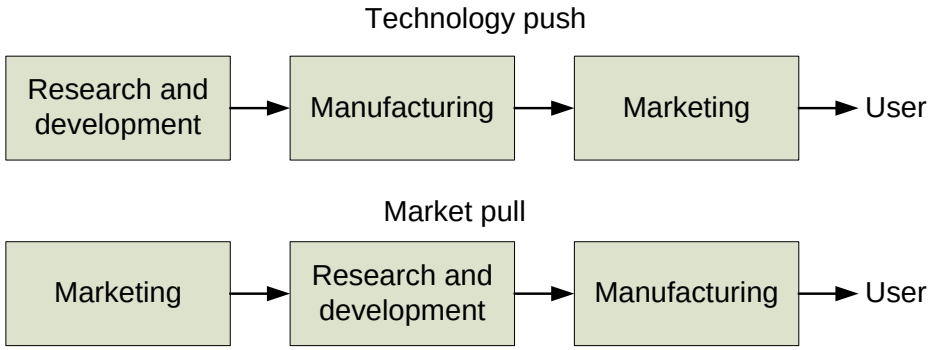

Figure 4 – Linear models of innovation (Trott 2005 pp.23)

In technology push model where it is assumed that scientists make unexpected discoveries. These discoveries are then applied into product ideas and engineers turn ideas to prototypes for testing. Manufacturing will produce products efficiently and finally marketing and sales promote products to potential customers.

Market pull model emphasizes role of a marketplace and customer needs. Marketing function is seen the initiator of new ideas resulting from the close interactions with the customer. These ideas, in turn, are conveyed to R&D for design and engineering and then to manufacturing for production.

3.1.2 Evolution to Coupling and Interactive Models

After linear models understanding of innovation evolved in 1980s to coupling model where emphasis was integrating R&D and marketing to work together. In 1990s focus shifted to interactive model (see *Figure 5* below) which develops coupling model further.

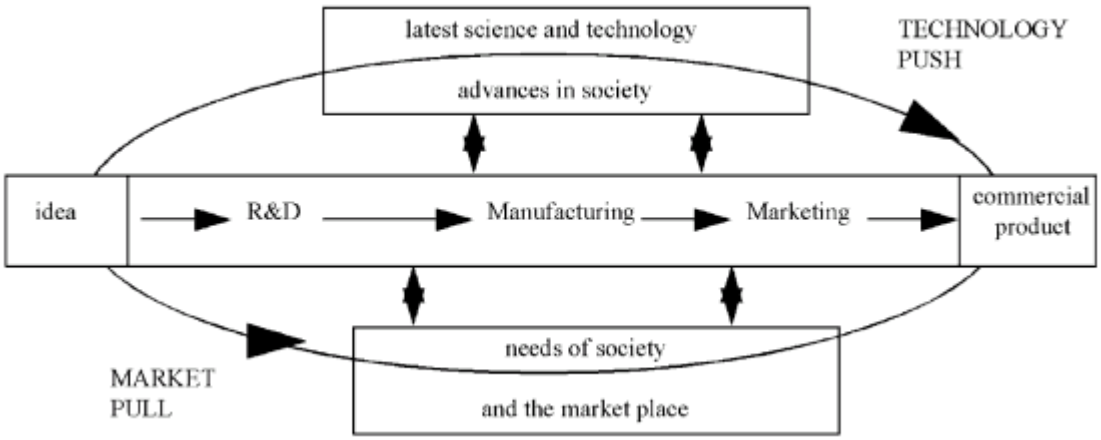

Figure 5 – Interactive model of innovation (Trott 2005 pp.24)

Interactive model emphasizes that innovation happens as a result of the interaction at the marketplace, the science based and the capabilities of the organization. Like with coupling model there is no explicit starting point. The overall innovation process can be thought of as a complex set of

communication paths over which knowledge is created and transferred. Organizations that are able to manage this process effectively will be successful at innovation. (Trott 2005 pp.24-25)

3.1.3 Open Model

In 2000s emphasis have shifted to open model i.e. knowledge accumulation and network linkages. Open model of innovation refers to principles coined by Henry W. Chesbrough (2003a). In his book Open Innovation (Chesborough 2003b pp. xxiv) states "*Open innovation is a paradigm that assumes that firms can and should use external ideas as well as internal ideas, and internal and external paths to market, as the firms look to advance their technology*". Opposite of open innovation is closed innovation where company itself is the only entity to create and profit from the innovation.

The boundaries between a firm and its environment have become more flexible as innovations can easily transfer in and out. The principal idea behind open innovation is that world is full of widely distributed knowledge and companies cannot afford to rely entirely on their own R&D. This means that in order to success in marketplace companies but should buy or license processes or inventions (i.e. patents) from other companies and internal inventions not being used in a firm's business should be taken outside the company (e.g., through licensing, joint ventures, spin-offs) (Chesbrough 2003a). Success and wide diffusion of open source software and Apple (store.apple.com), Google (www.google.com/apps) and Nokia's (www.ovi.com) application stores and ecosystems around them are good examples of open innovation.

Several factors that emerged have paved the way for open innovation:

- The increasing availability and mobility of skilled workers
- The growth of the venture capital market
- External options for ideas sitting on the shelf
- The increasing capability of external suppliers

These four factors have resulted in a new market of knowledge and this knowledge is not anymore proprietary to the company. It resides in employees, suppliers, customers, competitors, and universities. If companies do not use the knowledge they have, someone else will (Chesborough 2003a, 2003b).

3.2 Dominant Design Theory

Schumpeter's classical theory says that discontinuous innovation can trigger Creative Destruction. From Dominant Design theory (Anderson & Tushman 1990) point-of-view discontinuous evolution gives room for firms offering alternative and competing technology solutions to compete for future business. Dominant Design explains firms' success and failure from technology cycles and architecture perspective.

Main idea of Dominant Design theory is that technological discontinuities trigger a period of ferment that is closed by the emergence of a dominant design (see *Figure 6* below). This may happen due to incumbent technology discontinuity or due to breakthrough by superior competing technology.

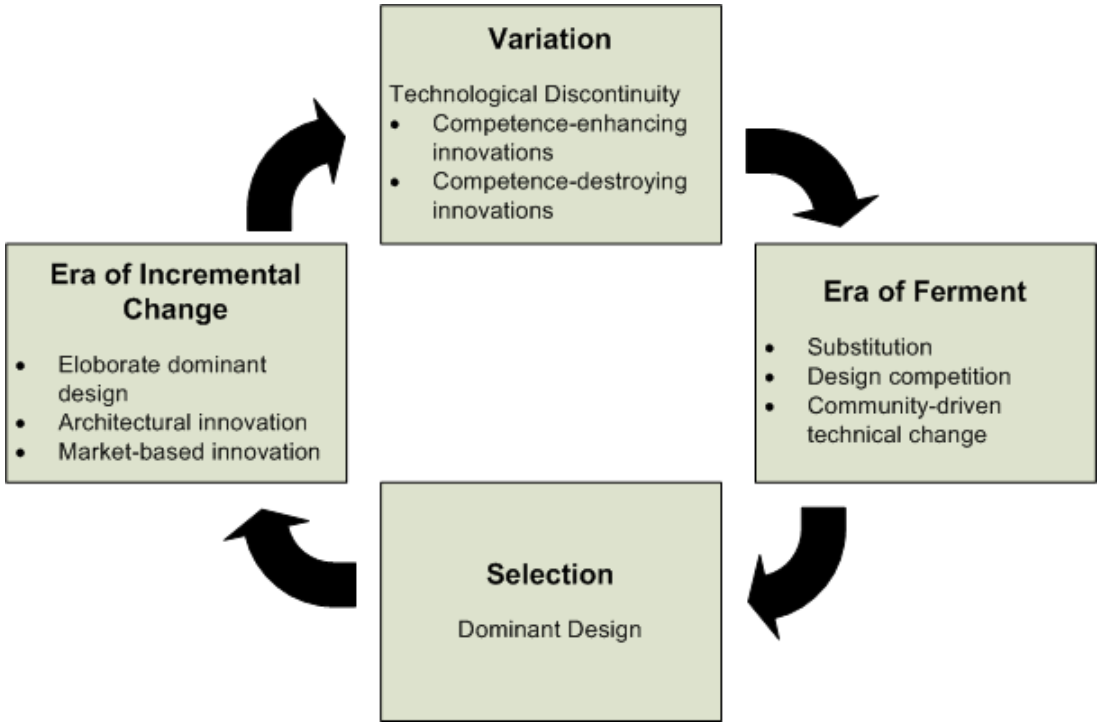

Figure 6 – Dominant Design theory (adapted from Rosenkopf & Tushman 1994)

Technological discontinuity means innovations that dramatically advance the customer's and potentially whole industry's price vs. performance frontier. After discontinuity a period of incremental technical change follows, which is in turn, broken by the next technological discontinuity. Some firms thrive during eras of ferment, other firms pro-actively destabilize their business during technological discontinues. Phases of Dominant Design theory are explained in *Table 6* below.

Table 6 - Phases of Dominant Design (adapted from Tushman & Smith 2002 pp.4)

Phase	Description
Technological Discontinuity	Industry faces a technological revolution which means for a enhancing or destroying phenomena from competence and business perspective. Many different innovations and technologies are created to solve similar problems.
Era of Ferment	Different innovations are competing. There will be technological substitutions and design competitions. Also different standardization bodies and forums try to develop and promote their own technologies.
Selection	Industry chooses the dominant design. This happens by two ways. Customers will choose their preferred technology and vendors choose to promote and produce their preferred technology. When both customers and vendors choose the same technology dominant design has been selected
Era of Incremental Change	There will be incremental development done on top of dominant design. Companies are competing by enhancing the dominant design and developing differentiating features based on architectural and market-based innovations.

3.3 Stage-Gate Model

Stage-Gate® model originates from the original research made by Richard Cooper (1979, 1980, 1985, 1994). Stage-gate model focuses on strict managerial process how new product ideas are screened, scoped, developed and launched.

3.3.1 Overview

Main idea of Stage-Gate model is that proper management of new product innovation process is essential for success (see *Figure 7* below). This means arranging necessary resources, talent, time and managerial framework for decision making but also allow creativity flourish in between the phases.

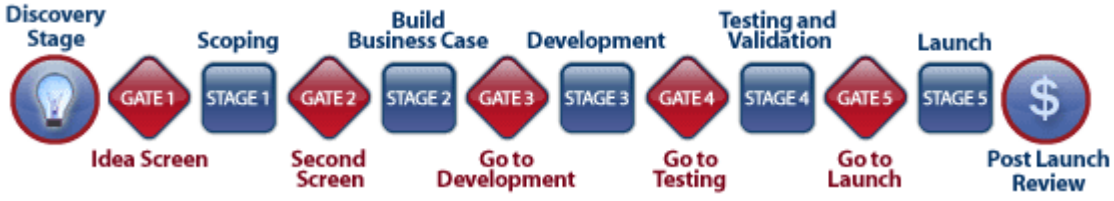

Figure 7 - Stage-Gate (Product Development Institute 2011)

In contrast to historical innovation models, which are conceptual in nature, Stage-Gate model acts as a managerial handbook with practical tools. Model focuses what organization has do in practice and when in order to launch successful products. Whereas original Stage-Gate model was strict sequential waterfall model the latest version, so called NextGen Stage-Gate® process (Cooper & Edgett 2008), incorporates features from open innovation (see section *3.1.3*) and lean/agile development principles (see section *5.4*). *Table 7* below explains the content of each phase.

Table 7 - Phases of Stage-Gate model (Product Development Institute 2011)

Stage 0 – Discovery	Activities designed to discover opportunities and to generate new product ideas
Stage 1 – Scoping	A quick and inexpensive assessment of the technical merits of the project and its market prospects
Stage 2 – Build Business Case	This is the critical homework stage - the one that makes or breaks the project. Technical, marketing and business feasibility are accessed resulting in a business case which has three main components: product and project definition; project justification; and project plan
Stage 3 – Development	Plans are translated into concrete deliverables. The actual design and development of the new product occurs, the manufacturing or operations plan is mapped out, the marketing launch and operating plans are developed, and the test plans for the next stage are defined.
Stage 4 – Testing and Validation	The purpose of this stage is to provide validation of the entire project: the product itself, the production/manufacturing process, customer acceptance, and the economics of the project
Stage 5 – Launch	Full commercialization of the product - the beginning of full production and commercial launch

3.3.2 Main Principles

NextGen Stage-Gate introduces set of principles for the process phases. In this study these principles are important as they will be analyzed and mapped with other principles introduced in this thesis. The following sections explain the main principles of Stage-Gate model (adapted from Cooper & Edgett 2008).

3.3.2.1 Customer Focused

One of the main reasons why new product development productivity is low is because of the lack of breakthrough new products with a "wow" factor in most

companies' development portfolios. Companies tend to spend most of their budget and talent to maintain the existing products or they invest to new product initiatives that do not differentiate or create real new value to customer.

Companies tend to forget that technology and features themselves are not important but the value generated by them is and the real key is to understand the customer.

Understanding the customer is not the same as voice of sales manager or voice of product manager. Companies should really invest to lead user analysis, focus groups, in-depth interviews, spend time with customer work etc. in order really to create deep understanding

3.3.2.2 Heavy Front-End Homework before Development

Due diligence in the early days of a new product project is essential. Right up-front homework pays for itself in terms of saving time and also higher success rates. According to research, consistently and across the board, high-productivity businesses excel here. Basic principle is to plan and understand the project, costs and risks and most importantly what is the value being sold to customers before committing time, money, talent and people to big project. As company should have multiple new product ideas to develop and launch selection process becomes crucial.

3.3.2.3 Spiral Development

In linear model product development starts with product definition based on information that might have been right at the time, and which they believed was accurate. Such a definition may even have been developed with solid process and really focusing to understand the customer. However, months then go by and soon project team realizes that original definition is not what customer is really needing because customer didn’t knew what she wanted. This causes slipping in schedules and budgets.

Thus, when it comes time for field trials or beta tests, everything has changed, and the product is not what the customer or market wanted at all. So launch is delayed and it's back to the drawing board. Indeed, unstable product specs and project scope creep are two of the main causes of long times to market, significant project delays, and even product failure. (Cooper 2001 ch. 2)

Interaction with users is important throughout development process as everything can't be understood and defined at the early stage. Please, refer to *Figure 8* below how Stage-Gate is applied for user interaction.

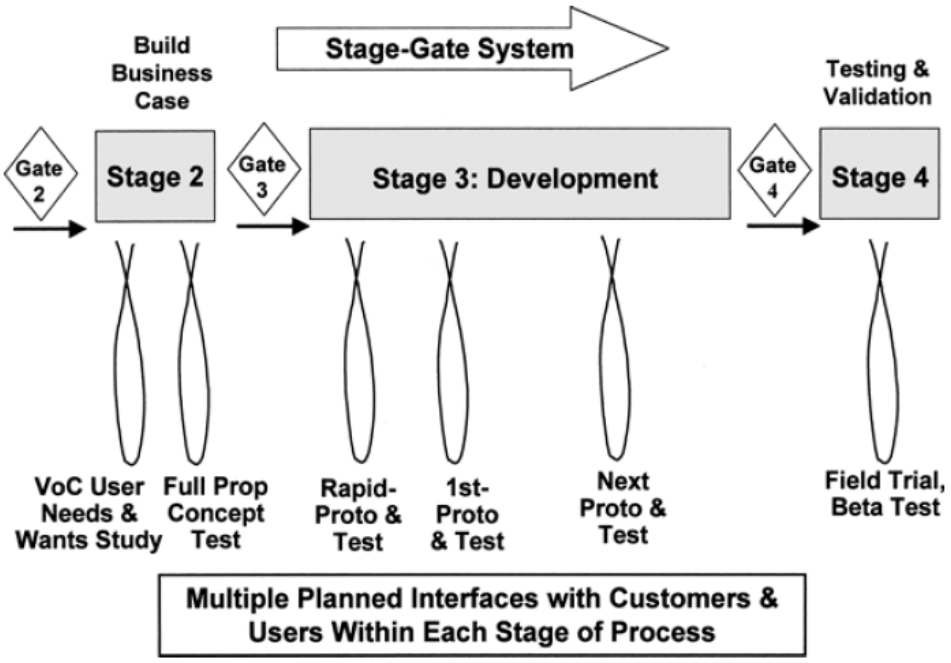

Figure 8 - User interaction in Stage-Gate model (Cooper & Edgett 2008 pp.10)

3.3.2.4 Holistic and Effective Use of Cross-Functional Teams

Product innovation is very much a business function - not an R&D activity - and team-based endeavor. And the core team, an effective cross-functional project team, is the primary key to cycle time reduction and to getting to market on time. In best-performing companies, effective cross-functional teams consist of key players drawn from different parts of the organization, and players are assigned so that it is clear who is on the team and who is not. The team is led by a carefully-selected leader or captain, driving his or her project down the field

to the goal line in entrepreneurial fashion, and also remaining leader from beginning to end.

3.3.2.5 Metrics

You cannot manage what you do not measure. Many businesses are guilty of not measuring their new product results—only 30 % of businesses measure the performance or outcomes of new product projects once launched. As a result, it's not clear whether a specific project was a success - met its profit targets or met its launch targets. Further, the post-launch review is one of the worst-rated activities in the entire innovation process, and is executed proficiently by only 22.1 percent of businesses. Without metrics in place, project teams cannot be held accountable for results, while learning and continuous improvements are next to impossible.

3.3.2.6 Focus and Effective Portfolio Management

Most businesses have too many development projects underway, and often the wrong ones. That is, they fail to focus, spreading their resources too thinly across too many initiatives, and their project choices result in the wrong mix and balance of development projects in the portfolio. Most companies do a poor job of ranking and prioritizing development projects; there are too many projects underway for the limited resources available, and portfolios contain too many low-value projects.

3.3.2.7 Lean, Scalable and Adaptable Process

Many businesses' idea-to-launch processes contain much bureaucracy, time wasters and make-work activities. By contrast, smart companies have made their new product development processes lean, removing waste and inefficiency at every opportunity. They have borrowed value stream analysis from lean manufacturing, and have applied it to their new product process.

3.4 Innovation Performance Differences

Whereas motivation for companies to innovate is clear, the actual innovation performances vary significantly between companies and between industries (see *Figure 9* below). This study focuses in IT where the difference between the best and the worst performer can be 30 folded when compared by the revenue generated by new products (Arthur D Little 2005).

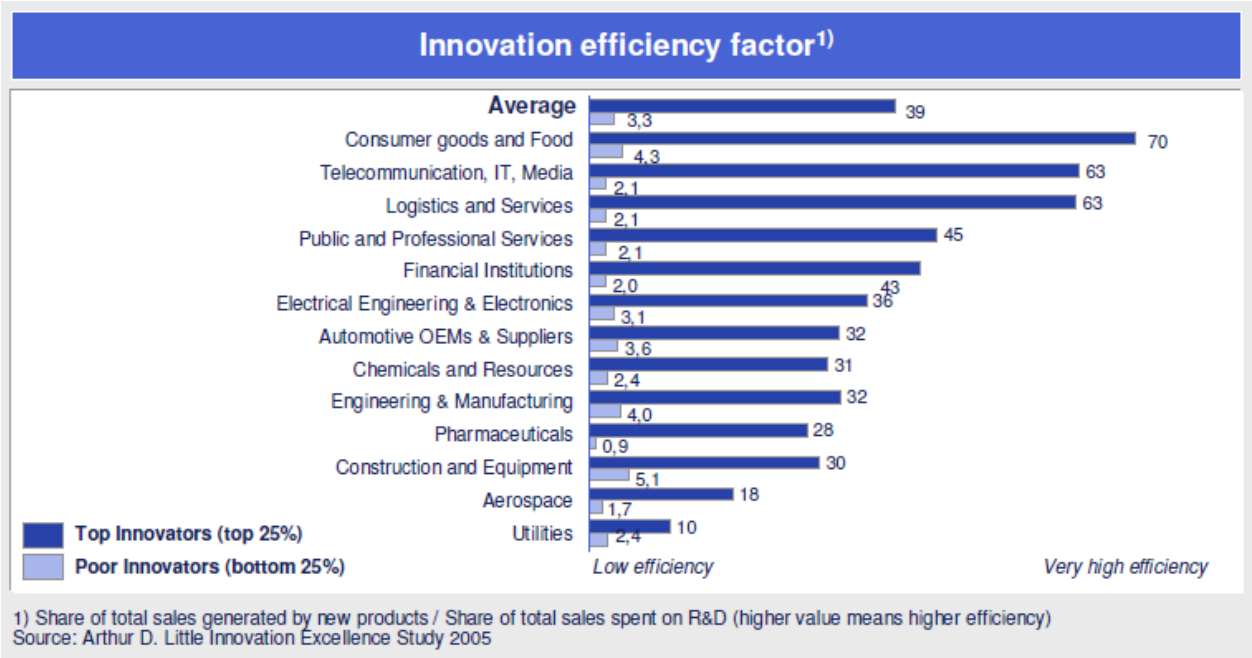

Figure 9 – Innovation efficiency in various industries (Arthur D Little 2005 pp.15)

Also Stage-Gate authors report that there is huge difference in between low and high new product innovation performer companies (Cooper, Edgett & Kleinschmidt (2001). This can be seen also how these companies utilize Stage-Gate principles in their daily work (see *Figure 10* below).

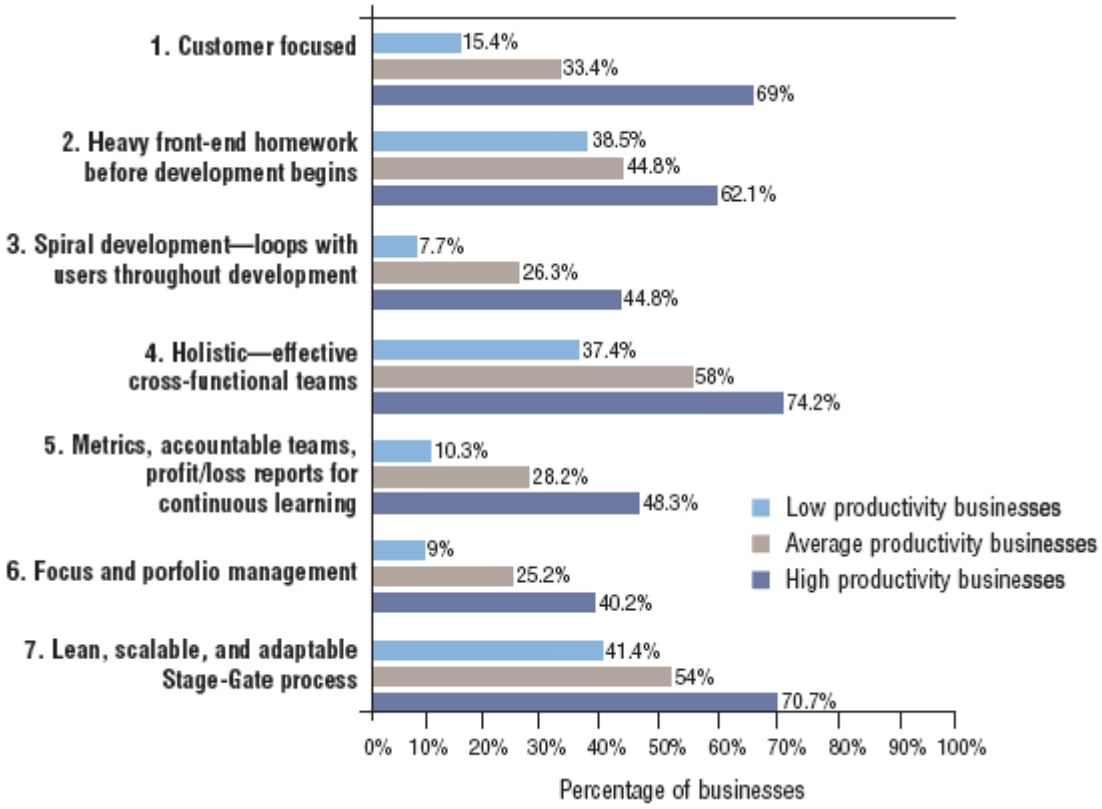

Figure 10 - Stage-Gate principles adoption difference among low, average and high performer companies (Cooper & Edgett 2008 pp.13)

3.5 Summary

Financially innovation is betting money for better return compared to alternatives with certain risk. As failure in new product innovation is high as 90% (Cooper 2001 ch. 2) expected return on investment is high. Schumpeter defined that target for innovation should be creative destruction meaning being ultimate winner in marketplace by reshaping the whole industry. Being the industry leader is best way to differentiate and win business as leadership is the most direct way to establish the credentials of product (Trout 2008 ch. 13).

Historically different innovation models are focusing value creation (i.e. developing technology, use value) and very little to value capture. Schumpeter's model is focusing to value capture as marketplace in terms of commercial exchange defines success.

Early technology push and market pull models were sequential assuming that superior invention or just doing what market demands is enough. Later innovation models evolved to include market research and interactive elements (i.e. customer testing). Recent addition is open model stating that not all creativity and capabilities are in the company and having network of partners helping with innovation is able to boost the results.

Stage-Gate is management model for doing practical innovation work. Whereas earlier models are conceptual Stage-Gate model describes how innovation should be managed in practice. *Table 8* below lists main features and principles of Stage-Gate model that are relevant in this study. Principles are also compared to historical innovation models.

Table 8 - Main principles of Stage-Gate model compared with historical innovation models

Principle	Description	Comparison to historical models (technology push, market pull, interactive mode)
Gatekeeping mechanism and heavy front-end work.	Stage-Gate has strict gatekeeping mechanism that controls advancing with the process. Main idea is to do each phase properly, do reviews and being able to kill non-lucrative projects and not committing all resources before proper analysis, business planning and preparations.	Historical models don't have defined gatekeeping mechanism. However, there is implicit assumption that planning including business case is done properly.
Project funnel / portfolio management	Stage-Gate model have principle that there should be many new product ideas (or inventions) for development and launch. During Stage-Gate process only few ideas reach to launch state whereas other ideas are eliminated because not being lucrative enough.	Historical models don't explicitly include portfolio management. Issue is that how company can kill new but non-lucrative product ideas if they don't have viable alternatives. Pressure from shareholders to innovate new products to achieve growth will not disappear. Companies typically have to have growth story even if they don't have good options in their portfolio.
Metrics	Innovation process performance should be measured in order to steer the process and for efficient learning.	Historical models don't have concept of measuring of innovation process performance.
Cross-functional teams	Teams participating to innovation process should be cross-functional and combining all relevant units in company. These are, for example, marketing, sales, R&D & customer services.	Technology push and market pull models didn't have cross-functional team concept whereas interactive model does.
Spiral development	Original Stage-Gate model didn't include spiral development principles and was purely sequential. NextGen Stage-Gate has spiral development principle to emphasize to have enough feedback from customers and end-users.	Technology push and market pull models don't have spiral development principle. Interactive model includes spiral principle.

4. Business to Business Sales

Sales view is typically excluded in academic innovation literature. This might be because of the lack of academic research tradition in the area. Sales view is very important because there the target is commercial transaction with the customer. Commercial transaction is actually the same where Schumpeter is focusing in Creative Destruction model. Creative Destruction happens only at the customer interface through series of commercial transactions i.e. company is constantly able to deliver much more value to customers compared to competition. Commercial transaction is also the source of company's profit and growth.

Professional sales literature focuses very heavily to practical business to business sales situations. This chapter draws key themes and principles which are important for the subject of this thesis.

4.1 Sales Process

Typical sales process to business customer includes several people in customer's organization. These are influencers, users, gatekeepers, champions and decision makers. From selling perspective the sales efforts, tactics and message are very different. For example, users are interested of the user interface and the technical features of the system but the decision maker is looking business benefits (i.e. more sales, cost savings, increased profit, optimized inventory etc.) and cares very little about technical features. For salesperson champions and decision makers are the most important ones.

Professional B2B sales literature focuses very heavily to message to decision maker. One of the main reasons for inefficient sales and unsuccessful technology products are bad and unstructured value propositions to decision maker i.e. how product helps business, what are the real business benefits instead of just having long list of technical features (Moore 1999 pp.148-161). Also just relying value propositions made by company's internal marketing or

product management unit is not enough as they are generic and fail to understand individual customers and competition there (Anderson, Narus & van Rossum 2006).

Traditional guideline for supplier to focus to customer needs is bit misleading as customers rarely can communicate what they need compared to what they want to achieve. There has been also change in past decade when customer demand and formality was strong (i.e. customers knew what they needed) compared to nowadays where customer expects companies to be experts to help with their challenges (Gova, Ghauri & Salle 2002, pp 48). For salesperson customer's targets are much more useful to be used for dialogue than needs (Shiffman 2004b pp 78-81). If customer can articulate their needs they already have decided what they want and there is much less room for salesperson to articulate alternative (and better) options.

Examples about customer targets and corresponding value propositions (Cheverton 2006 pp. 62):

- If customer is driven by operational excellence, focus propositions that impact on their supply chain, that reduce costs, and that improves efficiencies
- If customer is driven by product leadership, focus propositions that impact on their product, that improve quality, and that enhance their leading-edge ambitions
- If the customer is driven by customer intimacy, focus propositions that impact customers' customers, that increase their ability to respond, and that enhance their flexibility

The biggest obstacle to sales is not the competition, price or delivery schedules. It is the status-quo how customer is doing their business at the moment. Unless there is significant change of plan or problem customer will continue to do its business as earlier. Challenge for salesperson is to make customer to understand that using seller's technology will make customer's business better.

In practice this means that first challenge is to engage to dialogue with customer and to being able to ask questions to understand the customer's business target. After that the challenge is to present the solution effectively with clarity and impact so that customers understand the benefits and can act accordingly (managing this challenge see e.g., Schiffman 2007 pp. 4, Richardson 1998 pp. 1-13, Asher 2009 ch. 2, Konrath 2006 ch. 4, Moore 1999 pp. 148-161, Page 2002 ch. 6).

4.2 Crossing the Chasm

Geoffrey Moore's marketing management book Crossing the Chasm (1999) focuses to diffusion of innovations in high-tech industry in B2B sales and marketing context. Moore is book is based on his professional experience in Silicon Valley high-tech industry and he applies that to Rogers' (1962) theory of diffusion of innovations. The book and the model became very popular. Also the reasoning and principles of the model are very much in line with sales management literature (see *chapter 4*).

Moore's model focuses to disruptive and discontinuous innovations. Product enhancements that do not change customer's buying and usage pattern are excluded. Moore focuses on customers' buying usage pattern and what is the optimal sales strategy and value proposition in each phase of technology evolution and adoption lifecycle.

Main message in *Crossing the Chasm* is that reasons why different customer groups buy certain technology vary a lot and the order how technology is sold across the groups is fixed (Moore 1999, pp 30-55, see also *Figure 11* below). In Moore's model customer groups are divided to Innovators, Early Adopters, Early Majority, Late Majority, and Laggards. Every new technology in order to be commercially successful has to be first sold to Innovators and then to Early Adopters and so on.

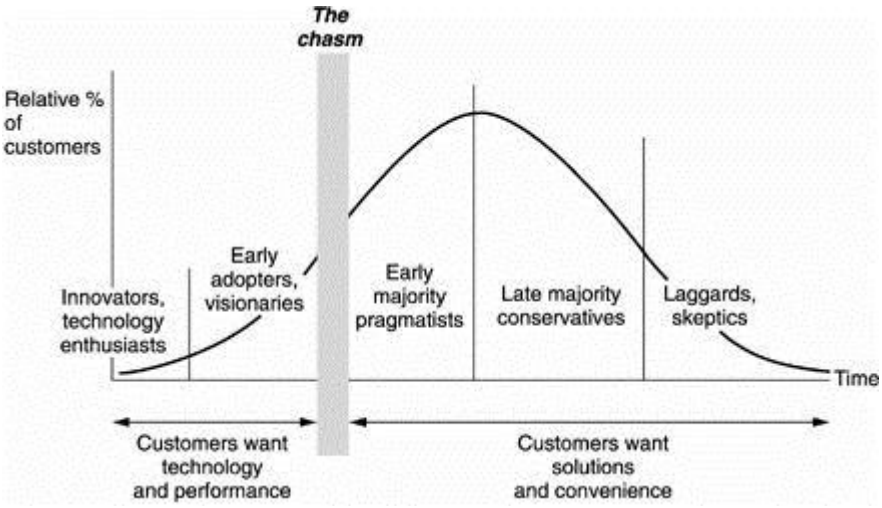

Figure 11 - Technology adoption lifecycle (Moore 1999 pp.12)

Biggest problem for any new innovation is the transition from Early Adopters to Early Majority as former ones are some sort of change agents wanting major leap in their business and get a jump on competition, the latter want to buy productivity improvement without experiencing the hassle and problems of productizing new technology. The following sections explain differences among the buying groups from sales perspective (adapted from Moore 1999).

4.2.1 Innovators

Innovators are technology enthusiasts and they appreciate technology for its own stake. In business, innovators are gatekeepers for any new technology. They are the ones who have the interest to learn about it and the ones everyone else deems competent to do the early evaluation. As such, they are the first key to any high-tech marketing effort.

As a buying population, or as key influencers in corporate buying decisions, innovators pose fewer requirements than any other buying group in technology adoption lifecycle. Innovators want the new stuff and they are willing to explore, learn and overcome the challenges of new technology. Innovators are the people who have energy to give feedback to vendors, make the products better and fix all bugs and issues.

Innovators want everything to be cheap. Sometimes this is a matter of budgets but it is more fundamentally a problem of perception - they think that everything should be free or available at cost, and they have no use for “added-value” argumentations. When successful, Innovators will be the ones who will praise and market new technology internally and externally and then becoming powerful ally when selling to Early Adopters.

4.2.2 Early Adopters

Early Adopters, or Visionaries as they are called in high-tech industry, are the people who have the insight to match emerging technology to a strategic opportunity. They have the temperament to translate that insight into a high-visibility, high-risk project, and the charisma to get the rest of their organization to buy into that project. Often working with budgets in the multiple millions, they represent a hidden source of venture capital that funds high-technology business.

Visionaries are not looking for an *improvement*; they are looking for fundamental breakthrough. Technology is important only insomuch as it promises to deliver on this dream. Visionaries drive the high-tech industry because they see the potential for an order-of-magnitude return on investment and willingly take high risks to pursue that goal. They know that they are going outside the mainstream, and they accept that as a part of the price to pay when trying to leapfrog the competition. Because Visionaries see such vast potential they are the least price-sensitive.

Visionaries are in a hurry but they like project based approach with milestones. As odds to fail are astronomical phasing becomes crucial.

The goal of each phase should be the following:

1. Is accomplishable by mere mortals working in earth time (i.e. is doable within reasonable effort in reasonable timeframe)
2. Provides the vendor with a marketable product

3. Provides the customer with concrete return on investment that can be celebrated as a major step forward

In sum visionaries represent an opportunity in product's lifecycle to generate burst of revenue and gain exceptional visibility. The opportunity comes with a price tag – a highly demanding customer who will seek to take company's priority and high-risk project that could end in disappointment for all. Visionaries are customers who give high-tech companies their first big break.

4.2.3 Early Majority

Early Majority, or pragmatists, will represent the bulk of market volume for any technology product. Company can succeed with Innovators and with Early Adopters but the majority of the money is with Early Majority and the ultimate success is with them.

From sales point-of-view pragmatists' goal is to make improvement – incremental, measurable, predictable process. They avoid risk and want that new technology has been trial and tested by Innovators and Early Adopters. They care about the company they are buying from, the quality of the product, the supporting infrastructure needed, and the reliability of the service they are going to get. Because pragmatists are in it for the long haul, and because they control the bulk of the dollars in the marketplace, the rewards building relationship with them are crucial.

Pragmatists are reasonably price-sensitive but they are willing to pay modest premium for top quality or special services. However, in the absence of any special differentiation they just want to have the best deal. Company successfully selling to pragmatists has to earn the reputation for quality and service i.e. being the obvious supplier of choice.

4.2.4 Late Majority

Late Majority, or conservatives, represent approximately one-third of total available customers within any given Technology Adoption Lifecycle. Conservatives in essence are against discontinuous innovation and they believe far more in tradition than in progress. And when they find something that works for them, they like to stick with it.

Conservatives often fear high-tech little bit and they tend to invest only at the end of technology lifecycle, when products are extremely mature and market-share competition is driving low prices and the products can be treated as commodities. Because conservatives are working the low-margin end of the market, where there is little motive from sales point-of-view to build a high-integrity relationship with the buyer, they often do get surprised with the problems of high-tech products.

In order high-tech business to be successful over the long term companies must establish reasonable basis to work with conservatives. Conservatives do not have high aspirations about their high-tech investments and hence will not support high-tech margins. Nonetheless, through sheer volume they can offer great rewards to the companies that serve them appropriately.

4.2.5 Laggards

Laggards, or the skeptics, do not participate in the high-tech marketplace except to block purchases. Skeptics inherently do not believe that high-tech improves productivity. The point is that cost-justification of high-tech purchases is a shaky venture at best. There is always the potential to return significant cost savings but it always depends on factors beyond the system itself. If high-tech marketers do not take responsibility for seeing that the whole product solution is being delivered, then they are giving skeptic and opening to block the sale. Most skeptics are struggling to point out is that new systems, for the most part, don’t deliver on the promises that were made at the time of their purchase.

The primary function of high-tech sales in relation to skeptics is to neutralize their influence. However, skeptics can teach what companies are doing wrong by pointing continually the discrepancies between the sales claims and the delivered product.

4.2.6 The Chasm

The chasm is the transition from the customers who are buying technology and performance to customers who are buying improvement and convenience. This transition is fundamental in many ways.

As focus is not the technology itself anymore high-tech marketer has to really understand all the benefits of its product not just high-level marketing wish-list. Many time companies fail to articulate the compelling application that provides the order-of-magnitude benefits.

Customers who are buying improvement and convenience want the whole product, not just part of it, including all the needed services and 3rd party technologies put together in order to create the solution in risk free form. Company just can't sell the vision without having the product. This means that the whole product has to be built and trial and tested first. Typically products in premarketing phase have still significant development hurdles to overcome.

Typical challenge is that company secures few pilot projects but the schedules and cost estimates tend to slip and the project priority in the customer's agenda falls and support is withdrawn. Despite the heavy development work not usable customer reference is gained.

Many times company providing high-tech product is new and it simply doesn't have the needed expertise to bring the product to the market. Company raises insufficient capital and hires inexperienced sales and marketing people. Company tries to sell the product through inappropriate channels and promotes it in wrong places and with wrong ways.

4.3 Reasons for Buying

Reasons why customers buy the solution are always individual and company cannot assume that second and third customers would buy a solution because of the same reasons as the first customers (e.g. Schiffman 2002 pp.115, Page 2002 ch. 5). This is analogous with technology enthusiast and visionary customers who buy systems with significantly different reasons compared to pragmatists and conservative customers (Moore 1999, refer also to *section 4.2.6*).

From B2B salesperson perspective companies buy because of two reasons (Schiffman 2004a pp. 21):

1. They either act to accomplish something that they themselves perceived as important before the salesperson showed up, or
2. They act in response to an opportunity the salesperson brings their attention

Value creation made by B2B sales can be understood to be tied to above reasons. With reason #1 sales' role is more to help and back-up the customer in their plans and winning the procurement process whereas reason #2 is something where sales have been able to communicate so much value (strong value proposition) that customer decides to break its current planning and operative status-quo and decides to buy supplier's solution. *Table 9* below summarizes how customer's perception of the problem and solution and supplier's sales mode, value proposition and technology maturity are compared to customer's buying reasons.

Table 9 - Comparison with customer's buying reasons

Item	#1 - supporting customers with their plans	#2 – breaking customer's status quo
Customer's awareness of problem and solution in hand	Customer is aware of the problem they want to solve and typically also aware of the available solutions.	Customer is less aware of the problem and the solution.
Supplier's value proposition for solving the problem	Customer's plans help supplier to form up its value proposition. Typically, supplier's value propositions are ranked against competition and are at least partly governed by formal procurement process. Value proposition focusing more to technical features and less to business benefits. Challenge is to differentiate with business level value proposition as customer is driving the process.	Value proposition has to be strong and concrete because it has to break customer's status quo. Less competition and formal procurement processes are in place to guide value proposition development. Value proposition has to have clear business focus.
Supplier's sales mode	Align with customer's plans and requirements. Try to understand customer better, find differentiator and communicate superiority against competition to enable more benefits to customer.	Focusing to deliver strong and concrete value proposition. After initial breakthrough fine-tune value proposition (solution) with customer. Little worry about competition.
Supplier's technology / whole product maturity	Varies, always compared to competition and alternatives. In more mature problem domains mature technology and whole product is expected.	Varies, however strong value proposition are easier to be built with mature technology and whole product.
Overall risk level	Varies, depends on problem domain maturity and competition.	Varies, however strong value proposition assumes that supplier is able to manage whole product related risks.

Whereas table above describes the buying reasons in B2B sales context, *Figure 12* below describes the nature of supplier's sales efforts in the context of technology evolution (Crossing the Chasm model). Figure explains that supplier's sales mode with innovator and early adopter customers is clearly cooperative to jointly develop technology and the solution whereas mainstream market customers are expecting supplier to provide whole solution. In other words, with innovators and early adopters supplier is creating use value (and

value proposition) together with customer and with mainstream market supplier is expected to have strong use value (value proposition) already available.

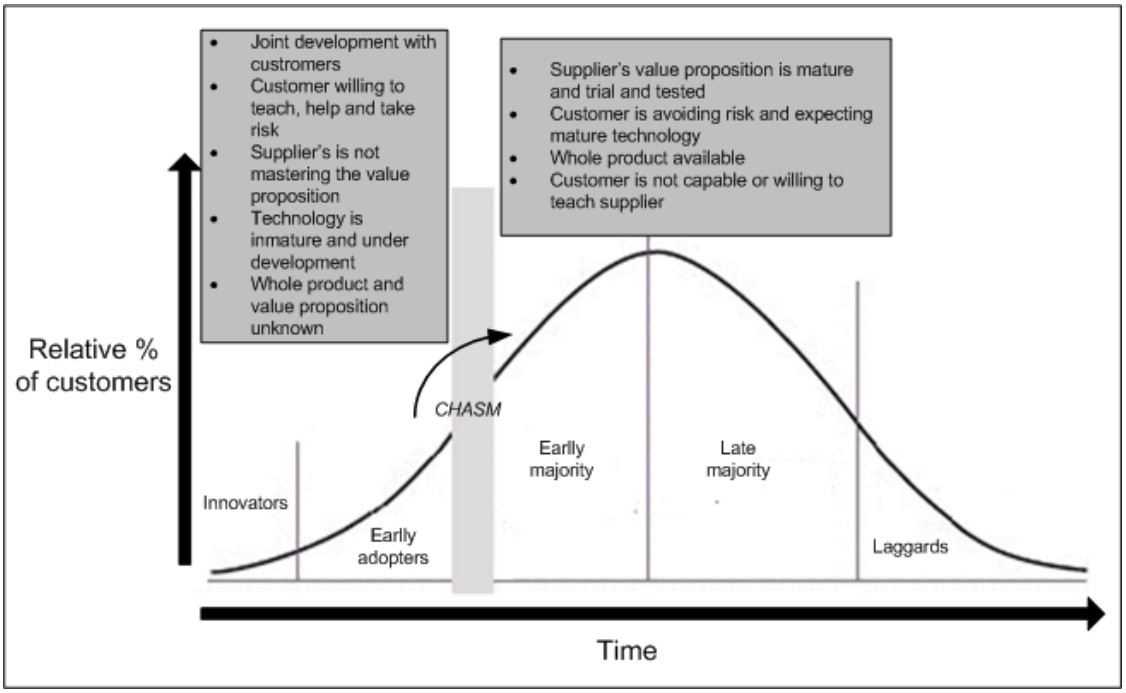

Figure 12 - Sales mode and value proposition development in Chasm model

In order to break customer's status quo supplier has to have strong value proposition. This means understanding really the business level benefits of offered technology, offering the whole project and communicating it efficiently. However, it may take years to learn and develop strong value proposition and that is really the challenge that suppliers have to win in order to enter the mainstream market. In Moore's model this challenge is defined as crossing the chasm.

5. Software Development

Software technology and development of software differ from physical goods in few fundamental ways. The supply side of software is characterized principally by high development, deployment and operational costs with insignificant replication and distribution costs compared to physical goods' high cost for all aspects (see *Table 10* below).

Table 10 - Supply cost elements of software products and material goods (Messerschmitt & Szyperski 2003 pp.345-346)

	High cost elements	Insignificant cost elements
Software product	• Development • Deployment • Operational • Maintenance (less)	• Replication • Distribution
Physical goods	• Development • Deployment • Operational • Maintenance • Replication • Distribution	• <none>

The most important distinction with physical goods and software is the inherent flexibility of software which can be transparently added to, repaired, and upgraded at any time, even as it is used (Messerschmitt & Szyperski 2003, pp. 23-24). Also flexibility in software is much desired and constantly developed feature in software technology and methodologies (Eden & Mens 2006).

On demand side software is very similar with physical goods and services. This is due to software's ultimate utility and value to customers is embodied in what it does similarly as with, for example, automobiles or traditional engineering artifacts. However, in B2B market software loses lot of its network effects due to high customer project specific costs compared to B2C market where consumer demand can be boosted by utilizing close-to-zero product distribution costs via internet. In B2B context typical software solution for customer include rather big

project phase for tailoring and deploying the software product. (Messerschmitt & Szyperski 2003, pp.311-317).

5.1 Software Evolution and Discontinuity

Rapid evolution of software is well known fact. Lehman and Belady (1985) have concluded that first principle of a software system is continuous change and constant development. This indicates software systems are under the ruthless pressure of change. When software system is used, it's needed to evolve along with the change of requirements and computing environments.

Software evolution can be characterized with continuous and discontinuous evolution (Aoyama 2001). Unlike natural evolution, software systems evolve on the hands of developers. Continuous and discontinuous evolution can be characterized as follows (see also *Figure 13* below)

- Continuous evolution: If required changes are small enough, under the level of tolerance to change, the systems can adopt them without major change at the hierarchical level.
- Discontinuous evolution: If required changes are beyond the level of tolerance to change, the systems need to re-engineer their architecture and evolve discontinuously. Such evolution may arise in the following two cases:
    1. A series of continuous evolutions reach the level of tolerance to evolution, and
    2. The required change is beyond the level of tolerance to change.

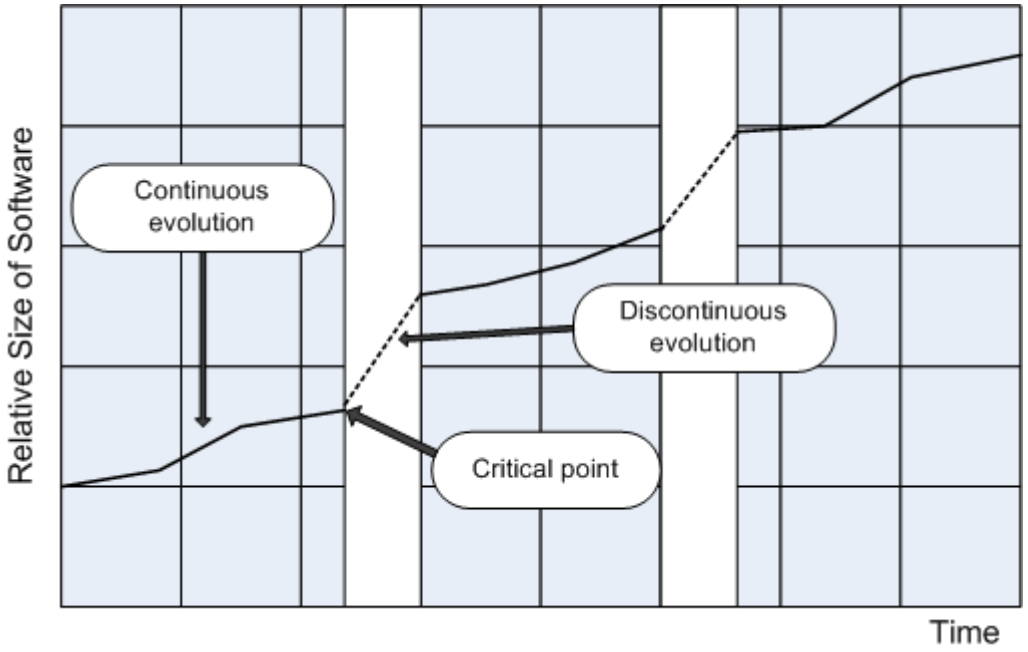

Figure 13 - Continuous and discontinuous software evolution (Aoyama 2001 pp.88)

5.2 Software Development Processes

In software development there are two distinct process models: waterfall process and agile process. This chapter explains the main principles and differences of these models as the models treat value creation and value capture very differently.

This study focuses on value creation and value capture of software development. Technological and pure technical project management aspects of waterfall and agile processes are well covered in general software engineering literature (for example, Boehm & Turner 2003 and Abrahamsson, Salo, Ronkainen & Warsta 2002).

Also, this study defines agile model broadly compared to usual definition in academia and in practice. Agile model in this study refers all process models where functionality is delivered in phases (i.e. iterations or sprints) and where content of each phase can be selected before the phase begins. For example, Rational Unified Process (IBM 2011) and Results Driven Incrementalism

(Fichman & Moses 1999) principles are considered agile in this study although in strictly speaking their practical project principles differ from pure agile (or Scrum) models. Reason for this broad definition is that this study focuses to the inputs and outputs of software process models and the terms and conditions of the projects and not to practical team work, development methods, communication or tools.

5.3 Waterfall (traditional) software process

Waterfall software development process is considered the traditional way to do software projects and it has its roots in US Navy experiments in computer programming (Benington 1983). Waterfall model (see *Figure 14* below) is applied from then-date de-facto engineering process models from machinery engineering, building construction and ship building.

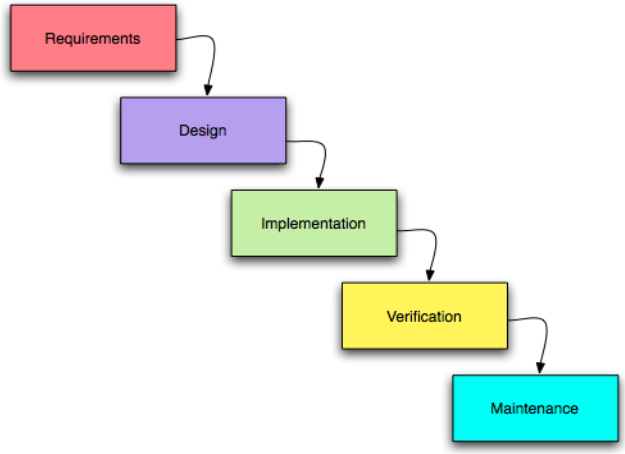

Figure 14 – Process in waterfall project (Smart 2007)

Main idea of waterfall software development model is series of sequential phases where the software is built. Driver for sequential model is that the software should be specified and designed as detailed as possibly in early phases order to avoid expensive and time-consuming changes and rework in later phases. Also it is believed that good quality is ensured by careful upfront

planning before the actual implementation work. *Table 11* below explains the process phases.

Table 11 - Phases in Waterfall model (adapted from Royce 1970)

Phase	Description
Requirements	Requirements state what the software should do. Typically requirements engineering starts from vague and abstract idea what the software should do. Good requirements state explicitly what the software should and what functionality is in or out of the scope. In customer and vendor relationship requirements document is many times considered legal document and part of the contract. Requirements document should answer all scope dispute questions.
Design	Design phase consist transforming the functionality described in requirements document to technical design and architecture of the software.
Implementation	Implementation is the part of the process where software engineers actually program the software code based on the design and requirements document.
Verification	Verification is the phase where the quality of the software is tested and verified against the requirements document.
Maintenance	When the software is ready it goes to maintenance where necessary defect fixes and updates to software are implemented.

5.4 Agile Software Process

Whereas waterfall software process aims to plan, design and write down in upfront what the software should do agile process model welcomes continuous changes to software design and functionality. Waterfall model is criticized heavily because it assumes that good upfront planning is possible and also as it can't manage changes the environment and business that will eventually happen during the lifecycle of the software project (Schwaber & Beedle 2002 ch. 1).

5.4.1 Values and Principles

The values and principles of agile software process are best described in Agile Manifesto by Agile Alliance (Agile Alliance 2001) (see *Table 12* and *Table 13* below). Agile Alliance is group of individual professionals in software industry who want to promote agile way of software development. Nowadays agile

software development is gaining more and more popularity and Agile Manifesto can be considered as a starting point of wider mainstream diffusion of using agile methods.

Table 12 - Agile manifesto (Agile Alliance 2001)

Agile values	Compared to waterfall software development
Individuals and interactions	Over processes and tool
Working software	Over comprehensive documentation
Customer collaboration	Over contract negotiation
Responding to change	Over following a plan

Table 13 - Agile principles (Agile Alliance 2001)

Principle	Description
1.	Highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2.	Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3.	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4.	Business people and developers must work together daily throughout the project.
5.	Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6.	The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7.	Working software is the primary measure of progress.
8.	Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9.	Continuous attention to technical excellence and good design enhances agility.
10.	Simplicity - the art of maximizing the amount of work not done - is essential.
11.	The best architectures, requirements, and designs emerge from self-organizing teams.
12.	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

.

5.4.2 Scrum Model

There are numerous agile software process models. For comparison please refer to Abrahamsson, Salo, Ronkainen & Warsta (2002). However, nowadays Scrum model and adaptations of that are the most popular (Eclipse Foundation 2010, Mountaingoat Software 2011).

Scrum process is iterative and incremental in nature, meaning that there will 2-4 long cycles (so called Sprints) where software is being developed and released

and design, implementation and verification phases will be done in each cycle (*Figure 15* below). Software features that are implemented in Sprints and prioritized based on their business value for customer.

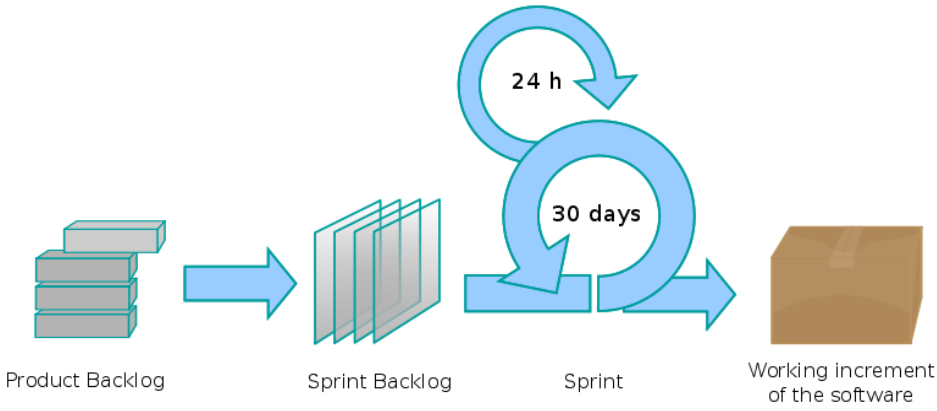

Figure 15 - Scrum process (Effective Agile 2009)

Main elements of Scrum process are listed in *Table 14* below. Please, note that Scrum process itself contains many other principles and elements but they are not relevant for this thesis – for more information please refer e.g. to Schwaber & Beedle 2002. Scrum model incorporates the phases from waterfall model but applies them differently.

Table 14 - Scrum elements and how they reflect with waterfall model

Scrum element	Description
Product backlog	Product backlog is a high-level feature list what the software should do. It is similar with Requirements in waterfall process but the idea is that list is prioritized, open and can change during the lifecycle of the project. In waterfall model Requirements document once approved is considered complete without priorities. Features in backlog with broad descriptions prioritized as an absolute ordering by business value. It is therefore the "What" that will be built, sorted by importance. It contains rough estimates of both business value and development effort.
Sprint and Sprint Backlog	Sprint typically a two to four week period where project team creates a potentially shippable product increment (for example, working and tested software). The set of features that go into a sprint come from the product backlog, Backlog items go into the sprint is called Sprint Backlog. Sprint Backlog is determined before

	the sprint. Software business owner informs project team of the items in the product backlog that he or she wants completed. The team then determines how much of this they can commit to complete during the next sprint, and records this in the Sprint Backlog. During a sprint, no one is allowed to change the sprint backlog, which means that the requirements are frozen for that sprint. Development is timeboxed such that the sprint must end on time; if requirements are not completed for any reason they are left out and returned to the product backlog. Sprint includes the design, implementation and verification phases for features listed in Sprint Backlog. After Sprint is completed, the team demonstrates how to use the software. End result of Sprint is working increment of software which includes the features listed in Sprint Backlog.
Working increment of software	Working increment of software is potentially shippable piece of software that can be released to customer. Business owner makes the decision about when to actually release any functionality or deliverable.

5.5 Conclusions

Waterfall and Agile process models are optimizing different aspects in project model. This can be seen from the definition of project (please, see *Table 15* below).

Table 15 – Definition of project in software process models

Model	Definition
Waterfall model	A project is an endeavour in which human, financial and material resources are organized in a novel way to undertake a unique scope of work, of given specification, within constraints of cost and time, so as to achieve beneficial change defined by quantitative and qualitative constraints. (Turner 1999 pp. 3-4)
Agile model	Five key business objectives of agile project management (Higsmith 2010a pp. 10-12): 1. Continuous innovation – to deliver of current customer requirements 2. Product adaptability – to deliver future customer requirements 3. Improved time-to-market – to meet market windows and improve return on investment (ROI) 4. People and process adaptability – to respond rapidly to product and business change. 5. Reliable results – to support business growth and profitability

In agile and waterfall models there are lot of other differences that are focusing actual software development practices, teamwork, collaboration and communication. However, they are not in the focus of this study.

Waterfall model assumes that detailed scope and all creativity happen at the beginning in Requirements phase and written in (potentially legally binding) documentation. Based on requirements documentation and planning project objectives can be delivered according to agreed cost, schedule and quality. Essence of success in waterfall model is that early stage of the project can be done with good quality. In practice this means that requirements can be understood in detail and written down exactly and unambiguously and the planning (including effort estimates and schedules) can be done realistically taking into account available resources and capabilities.

Agile model assumes that initial requirements phase and planning can’t be done with enough good quality. This is due to various reasons like nobody knows all the requirements at the beginning well enough, nobody can write requirements down or writing detailed requirement takes too much effort and time.

Also, requirements will change during the project and it is not important to deliver all requirements but only those that are most important. As requirements will be unclear and incomplete there is no point to estimate project effort and schedule based on inputs that are not with good quality. In practice there are no good tools to make effort estimates at the beginning. Tools using historical data would be nice but there aren’t any available as every project is unique. Quality suffers as supplier is in practice forced to cut the corners in order to maintain the schedule and in reality resources can’t be really committed to project due to conflicting priorities.

Agile model states that it is better deliver partial software frequently than expecting one big (potentially unsuccessful) full delivery at the end of the project. Waterfall approach assumes (implicitly) that completing the requirements phase reduces the most risk – an unlikely scenario for most projects

As Jim Higsmith (2010a, pp 17) puts this:

> *"A traditional project manager focuses on following the plan with minimal changes, whereas an agile leader focuses adapting successfully to inevitable changes"*

5.5.1 Project Triangle vs. Agile Triangle

Project triangle (or Iron Triangle) is old concept to describe the nature of projects and project management (see *Figure 16*). Project triangle describes well the philosophy of traditional project management (Turner 1999 pp. 9). Idea is that scope is agreed (contracted) with certain cost and schedule with agreed quality (note, good quality is usually assumed coming from professional project management, work practices, standards and regulation). Although project-triangle has been criticized it still continues to be the preferred success criteria (Atkinson 1999).

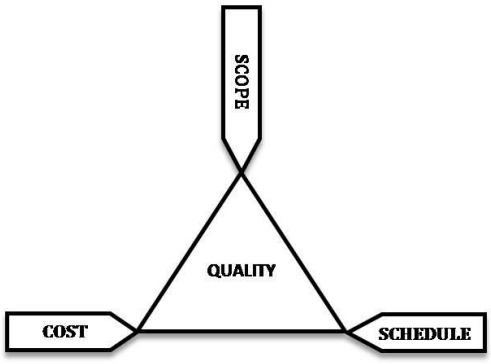

Figure 16 - Project Triangle (Rodney 1999 pp. 9)

From customer point-of-view supplier takes the risk to deliver project with agreed scope, cost and schedule. Supplier has to incorporate the risk in the price and schedule by using its own expertise. If the project will be done in-house then internal customer expects the same from internal supplier.

Agile model sees project triangle very differently and has created own definition – Agile triangle (Higsmith 2010a pp. 329, see also *Figure 17* below). Agile triangle philosophy is that development team (supplier) will build software in

phases where content of the each phase can be selected as suits best to customer with good quality within certain constraints. Constraints are coming from traditional project triangle where cost refers the running cost of development team, schedule refers to overall available time for customer and scope refers to the overall business target of customer.

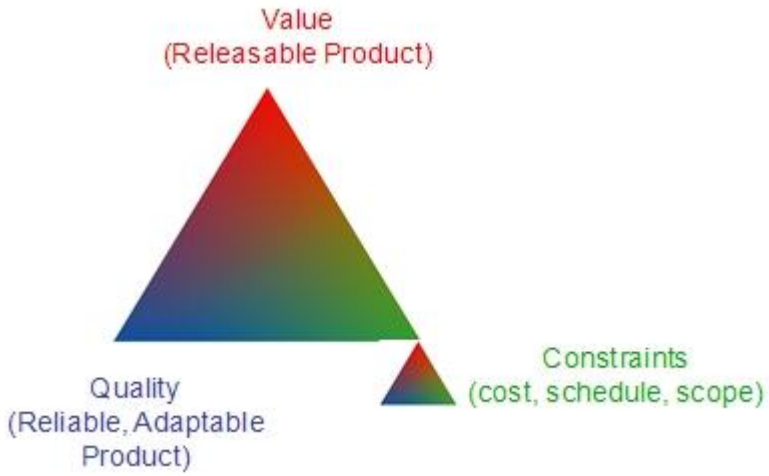

Figure 17 - Agile Triangle (Higsmith 2010b)

One of the main challenges of Agile triangle is that the development team (supplier) doesn't commit to scope, cost and schedule. Although agile is seen superior model nowadays to implement software (Ambysoft 2008, Ambysoft 2010) the problem of cost and schedule, or more precisely Return on Investment (ROI) will not disappear from top management's agenda.

Investment decision point-of-view ROI with schedule and risk associated to achieving ROI are fundamental. Money is spent in order the get agreed benefits in agreed time. Traditional project model fits nicely in to this thinking as supplier commits time, schedule and cost in contract whereas agile model doesn't.

Problem many times is that when organizations, who have earlier worked with traditional waterfall model, move to agile and stop committing to any schedule, cost and scope. However, financially and investment planning perspective someone has to commit. In practice agile model needs additional organization

that can manage and calculate risk in terms of ROI in order to be compliant with investment thinking.

Investment point-of-view there is roughly two kinds of projects (Higsmith 2010a pp. 309):

- Known problem – known solution project
    - Project scope, schedule and cost can be determined because problem domain and solution is known
    - Type of project where risk the contractual risk of schedule, scope and cost can be transferred supplier (or supplier can reasonably take and price that risk)
    - This kind of project can be contracted with traditional model.
- Exploration project
    - Project is unknown in many ways from problem and solution point-of-views. Project target is more an idea or dream and no known solution is available.
    - Scope, schedule and cost risk can't be transferred reasonably to supplier as the risk margin in price and schedule will skyrocket.
    - This kind of project is about learning and fits well with agile model

5.5.2 Value Creation and Learning

Waterfall and agile project models differ substantially how they create value (see *Figure 18* below). Waterfall model delivers value when project is over in single delivery. Agile model delivers value in phases and the content of each phase is decided before the phase starts.

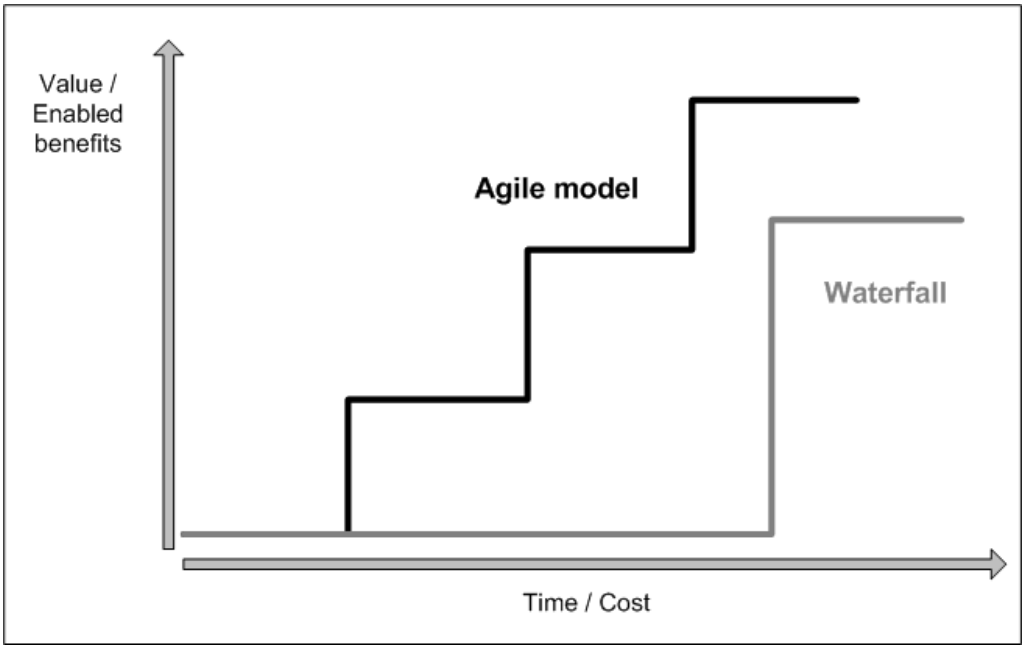

Figure 18 – Agile vs. waterfall value creation (adapted from Fichman & Moses 1999 pp.42)

One should note, that although agile model assumes that each phase (iteration) delivers value for customer it only has use value to customer if customer is able to use the delivered functionality. Challenge is, especially in larger projects, that agile phases do not deliver any meaningful use value for customer as deliverables are too tiny to enable any improvements to customer's operations or business. It might require several or dozens of agile phases to deliver meaningful improvement to customer processes. This same challenge is in company internal R&D where most of the releases doesn't bring any sellable or marketable benefit to company (Higsmith 2010a pp. 29, 36).

Also, the figure above doesn't explain the innovation and learning in detail from value creation point-of-view. Agile model introduces feedback loop, which allows learning during the project. One should note that learning has been identified earlier one of the fundamental elements of R&D (Cohen & Levinthal 1989).

Whereas waterfall model sees project as a vehicle to deliver agreed scope agile model sees project as a vehicle to learn and to adjust the project scope optimally (see *Figure 19* below). In waterfall all the learning happens in requirements phase and that phase can take as much calendar time as the project implementation.

Agile model enables project to start with only vague definition of scope and delivers value in short cycles. As already mentioned earlier in this chapter, the idea of waterfall process is to protect the scope and introducing changes to scope should be avoided because that will jeopardize the schedule and cost.

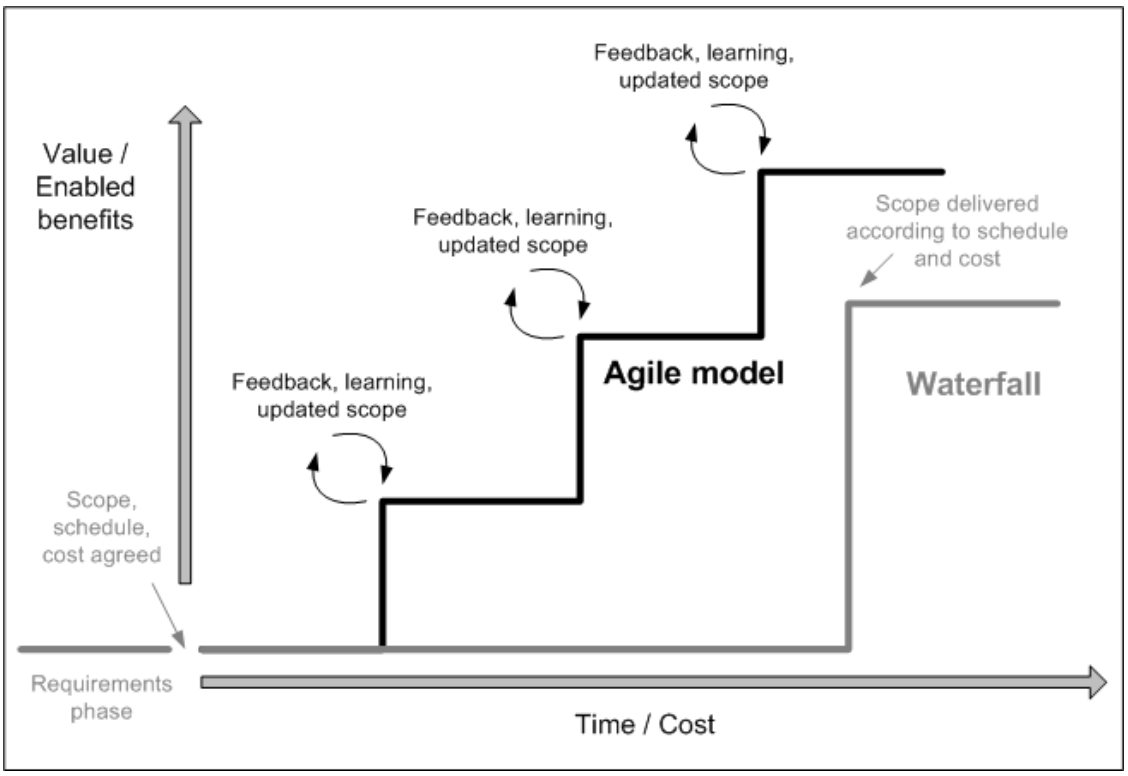

Figure 19 - Learning in waterfall and agile models

5.5.3 Project Risk

Project risk is managed differently in waterfall and agile models. In waterfall model project success can be determined only at the point of delivery (unless delays or invalid scope is reported during the project). In agile model, due to the

nature of deliver functionality in phases, risk is reduced significantly during the project as customer can evaluate the project progress in each phase and adjust the project scope and planning accordingly (see *Figure 20* below).

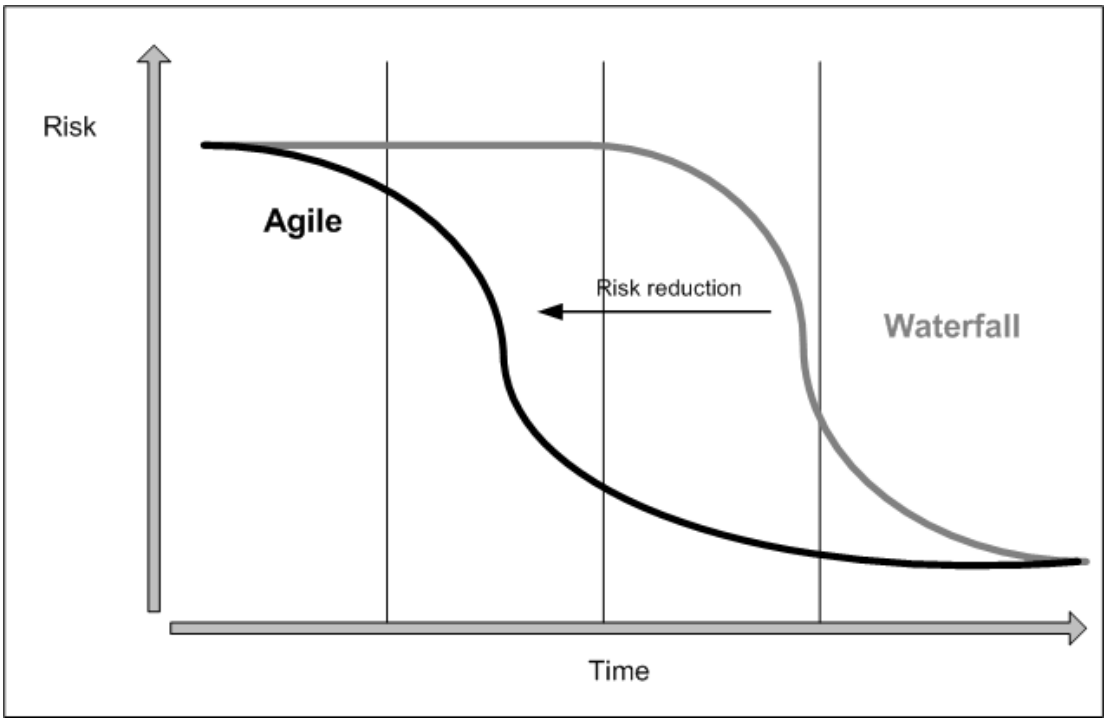

Figure 20 - Risk reduction in agile vs. waterfall process (adapted from Kroll 2005)

However, generally accepted model for agile's risk reduction mechanism (Kroll 2005) in not entirely true. Main issue is that in waterfall model development team (supplier) agrees to take risk to deliver agreed scope in agreed time and with cost whereas in agile model this risk is transferred to customer.

Contractually agile supplier typically commits to supply only professional team with time and material pricing and leaves scope, schedule and cost risk to customer. Although in waterfall model customer at the end carries the ultimate risk in agile model risk sharing between customer and supplier is not possible with the same way as in waterfall model. However, as supplier doesn't price risk in the project price customer can be expected to pay less.

5.5.4 Model Applicability with Big Projects

Waterfall and agile process applicability with big projects has been active area in management literature. However, the focus have been more how to manage big agile projects from communications, process and collaboration point-of-view and the value of agile outputs in big projects has only recently gained focus.

One of challenges in agile development is the fundamental principle that each iteration (phase, sprint) output should bring value to customer. However, if the customer is not able to benefit the small functional increments delivered by agile team (3-10 developers) agile model fails to deliver any business benefit. Original agile models, like XP and Scrum, didn't put focus to long term planning as their focus was to adapt always to current situation by stating that scope will nevertheless change. This is one the reasons why lack of long term planning to deliver real use value to customer is endemic in agile community (Higsmith 2010a pp. 157). On contrary, waterfall model's planning and project cycle aims to deliver business benefit when project is ready and is aligned better with top management's agenda for investment decisions.

Project scope creep (see e.g. Lehmann 2006, Cutting 2007) is another issue in projects general. As agile principle is to adapt to situation, it always aims to fill the development iteration with features that are considered important. However, in waterfall model idea is to protect the project scope and add as little as possible new features. Challenge is that with agile teams and business owners (typically product managers) that, although their intentions are good, they tend always fill the development cycle with things to be done (as mentioned in *section 5.1*, there is always need to develop the software). Polishing existing software by making it better improving quality, performance or adding features which don't bring value is called gold plating (see e.g. Atwood 2004). Gold plating is not issue only with agile development as same happens in waterfall models. However, agile model is very prawn to gold plating as changes to scope do not have to be approved by change management procedure.

Only recently, agile community have identified above mentioned challenges (i.e. short vs. long term planning issues and the importance of scope protecting) that there should be separate longer term release planning and management procedure, project portfolio management, that controls agile projects and development teams (Higsmith 2010a ch. 8). Waterfall model focuses to deliver value (scope) against certain cost and schedule and has built-in mechanism to protect the scope. Pure agile model is lacking both mechanisms but on the contrary waterfall model rarely succeeds to define the project scope satisfactory and estimate project effort and cost.

5.5.5 Value Capture

Section 2.1 defined that value capture happens when sales realizes. The amount of captured value is defined as exchange value i.e. price paid for the use value (product). To be more precisely, exchange value is bound to invoicing schedule that is defined in contract. In in-house development, the development team itself doesn't capture any value as no money is exchanged.

Waterfall contracts are usually constructed so that customers will pay when supplier reaches certain milestones and whole contract value is invoiced in phases (i.e. payments to supplier are hold until the milestone is delivered and supplier carries the risk to deliver the milestone). The nature of agile process is that supplier doesn't carry traditional milestone risk as pricing typically follows time and material principle and invoiced monthly or iteration basis. As iteration is short and the length is fixed (2-4 weeks) and the supplier selects only those deliverables in the iteration that really can be delivered, supplier's risk is minimal and supplier gets payments regardless what it delivers to customer. See *Table 16* and *Figure 21* below for comparison.

Table 16 - Waterfall vs agile value capture and payment milestones

Model	Typical value capture / payment milestones
Waterfall	Payment milestones; 1. Contract signed • Parties have signed the contract and purchase order is received from customer 2. Detailed design ready • Supplier states that detailed design of the software is ready 3. Ready for acceptance • Supplier delivers software to customer's acceptance testing 4. Accepted • Customer has accepted the software, software goes live and warranty period starts 5. End of warranty • Warranty period ends and all supplier's obligations are fulfilled unless there is separate support and maintenance contract
Agile	Payment milestones; 1. Iteration done / 1st month 2. Iteration done / 2nd month 3. Iteration done / 3rd month 4. Iteration done / 4th month

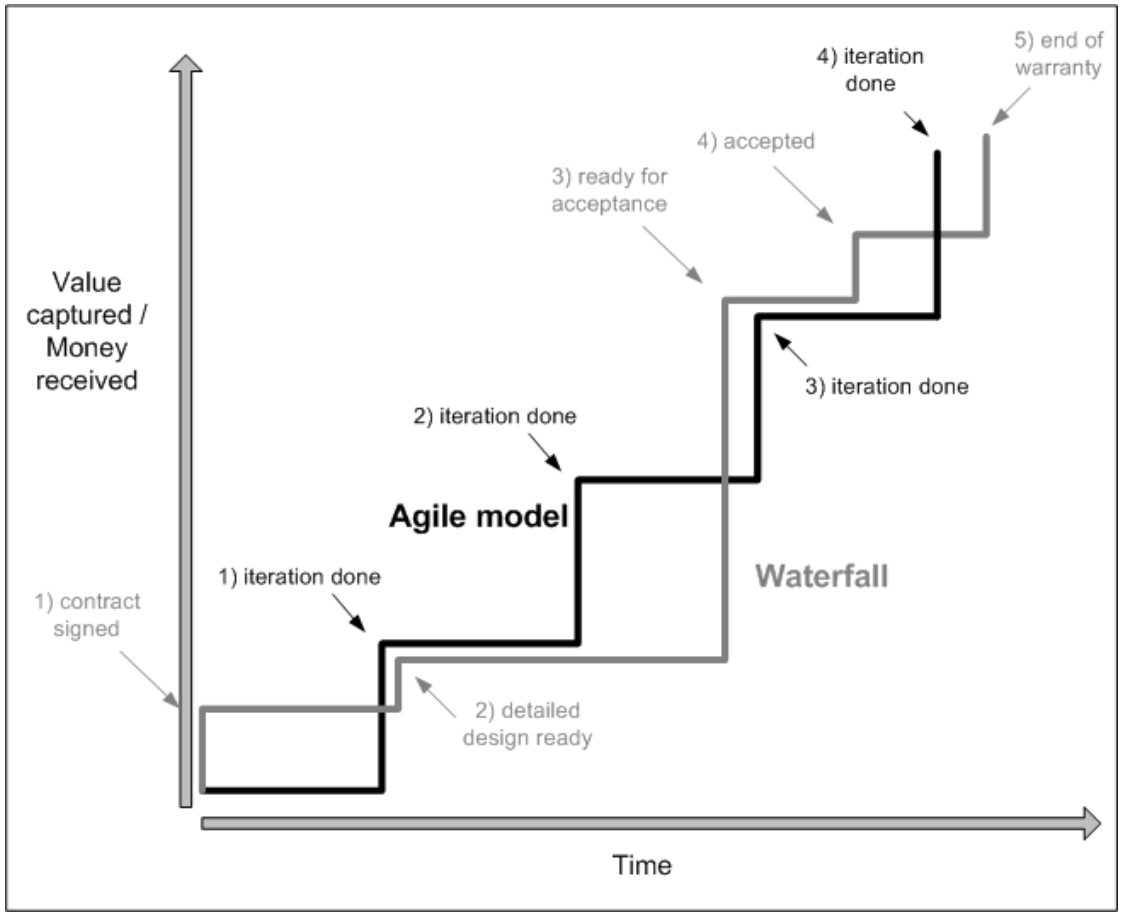

Figure 21 - Waterfall vs. agile value capture

In addition to price, bargaining power between customer and supplier in waterfall model typically goes so that customer tends to negotiate payment milestones as much as possible to the end of the project or even to the end of warranty period and also to increase the payment term. In agile model payments are monthly or iteration based and payment term is only thing to negotiate as supplier's pricing follows time and material and customer is responsible of overall scope and schedule.

In principle, if project is large enough the value capture mechanism of waterfall and agile models tend to converge as customers are expecting enabled business benefits in large scale instead of small functional increments delivered by agile iterations that don't bring business value. However, agile model combined with long term release planning mechanism is able to learn and adapt

as business situations evolve whereas waterfall model assumes that requirements phase captures all necessary elements and features of the project and is really able to minimize the risk.

6. Analysis

This chapter discusses the principles and findings presented earlier in this study. *Table 17* below summarizes the items that are compared and analyzed in the following sections.

Table 17 - Summary of analysis chapter

Items	Summary
Up-front planning in Agile process and in Stage-Gate.	Agile process' principle to avoid up-front planning is in contradiction with Stage-Gate model's emphasise to focus on up-front planning.
Value creation in Agile process and in Stage-Gate.	Both Agile and Stage-Gate processes define value creation differently.
Value creation in software processes (waterfall and agile)	Value creation of software process (both agile and waterfall) is dependent of what customer is buying.
Focus area of Stage-Gate and Chasm models	Stage-Gate focuses to launch products for the first innovative customers whereas Chasm model focuses to mainstream markets. This conflict is analysed.
Focus of Stage-Gate and Chasm models are analysed with Creative Destruction model.	Creative Destruction model is linked together with Stage-Gate and Chasm models.
Stage-Gate, Chasm and Creative Destruction models are linked with Dominant Design theory.	Dominant Design theory is linked with Stage-Gate, Chasm and Creative Destruction models. Stage-Gate model doesn't take dominant design principle in to account.
Role of sales channel analysed with literature presented in this thesis.	Interestingly, the sales channel is in absence in the most of the models covered in this thesis. Models many times assume that sales channel exists and can sell the products and the sales channel doesn't bring any requirements in to processes.
Waste in product development and software process in analysed	How software process models create waste and how they deal with it.

6.1 Agile principle vs. Stage-Gate big upfront planning

As explained in *section 5.4* one of the main principles of agile software development is to avoid big upfront planning. This is justified by stating that it is much wiser to adapt to changing circumstances and to new business requirements than trying to plan in detail beforehand.

Stage-Gate model (*section 3.2*) has principle of heavy upfront planning to ensure right decision making and success of the launch. At surface, these two

models seem to be fundamentally in contradiction with their principles but further analysis sorts the conflicts.

6.1.1 Analysis of Differences in Agile and Stage-Gate Models

Agile principles are to ensure successful software development. This also means that agile software process covers only software development and not the whole innovation process (e.g. idea, development and commercialization). Stage-Gate model focuses the whole innovation process.

Pure agile models don't have long term planning principle. Meaning that agile model tries to adapt to changing situations and not trying to ensure delivering certain fixed set of functionality in agreed time-frame. Agile model leaves feature decision making to product owner (typically product management or marketing function).

Stage-Gate model assumes that detailed product feature planning upfront is possible. Agile model states that this is usually impossible.

Stage-Gate model has built-in budget control mechanism (i.e. product should be ready after certain investment). Agile model doesn't have budget control.

Agile model doesn't give good guidance to product owner how to decide product features. Recent agile management literature (Galen 2009 ch. 7, Forss 2009 pp55) suggest finding and calculating business value for each feature but also confirms the huge practical challenges in doing it.

As discussed in *Chapter 4 Business to Business Sales* corporate decision makers do not buy technical features. They buy business solution. Agile model is only able to deliver technical features. It is the role of product owner (marketing) to translate technical features to business benefits.

As discussed in *section 5.5.1* Agile model is good for learning and exploration. Pure Stage-Gate model assumes known-problem known-solution project which is rarely true in new product development initiatives.

6.1.2 Combining Agile and Stage-Gate

To succeed with agile model and Stage-Gate model as both of them can be considered best-of-breed management models one has to distinguish the differences and understand the consequences.

Stage-Gate model focuses creating solution with business benefits including the governance for the whole project. Agile model focuses to create good software, in other words this means technical product with technical features, without whole project governance model

Stage-Gate is the management model for marketing and product management (although teams should be cross-functional). Agile model focuses on software development and leaves feature prioritization to product owner.

Big mistake is to assume that agile model's adaptability principle to avoid upfront and long-term planning will give success on marketing and new product process level. Agile model needs strong marketing (product management) to be successful as marketing controls agile development. For example, author's own experience is that product and marketing managers who were originally agile developers heavily underestimate the need of up-front planning causing lot of performance issues in new product development.

6.2 Agile vs. Stage-Gate value creation

Value creation happens in different levels in Agile and Stage-Gate models. Agile model's value creation is two-folded. First, each iteration creates new set of functionality and secondly each iteration creates possibility to have feedback and to learn. Stage-Gate model's value creation is also two-folded in addition to the learning and creating the product itself. First, Stage-Gate model develops

(business level) value proposition and secondly it creates understanding and plan for business potential in company level. *Figure 22* below illustrates value creation of technical product by agile process and value proposition creation by Stage-Gate process.

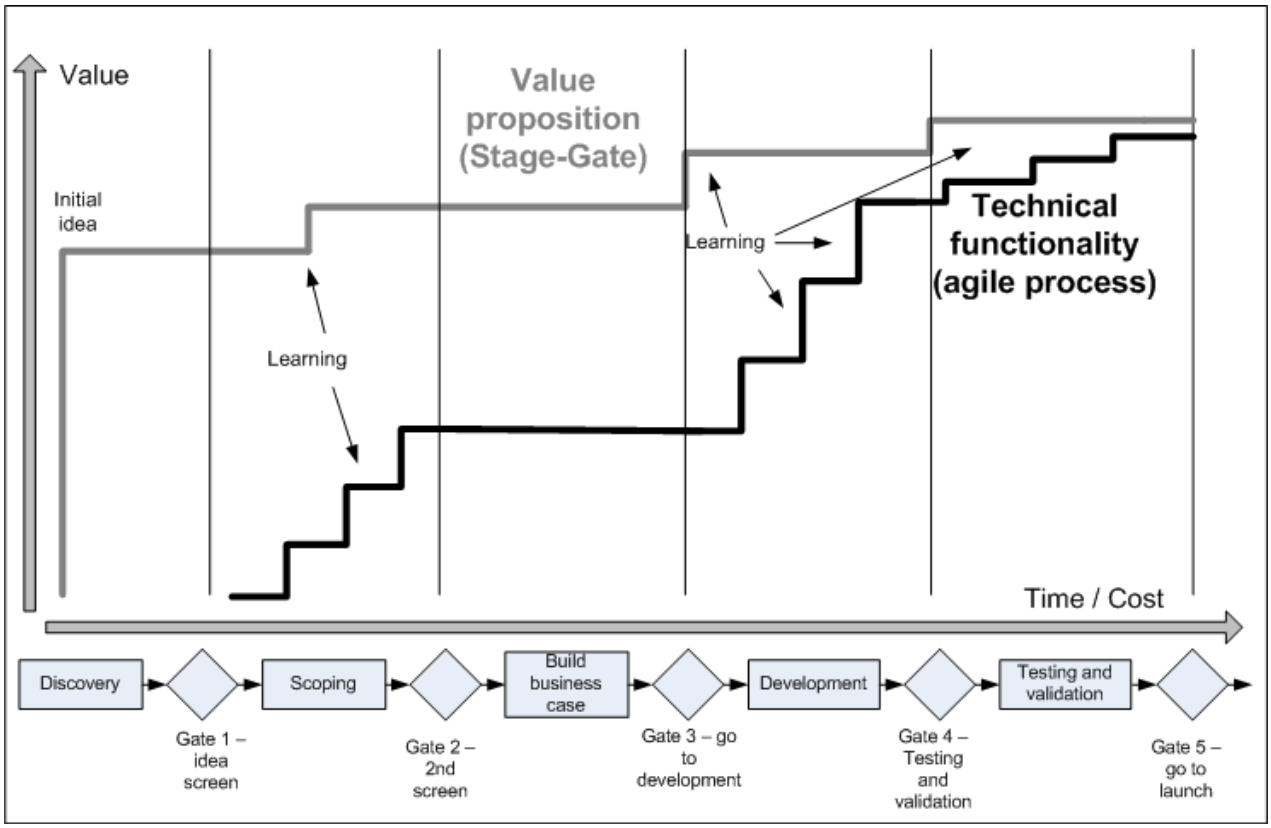

Figure 22 - Agile and Stage-Gate value creation (simplified)

One should note that conceptually creation of value proposition happens earlier compared to Agile model's technical functionality value creation. In Stage-Gate model initial idea after scoping should already include the main elements of value proposition as the value proposition is tested with potential customers. Rest of the Stage-Gate process should only fine-tune the value proposition. However, the technical product is mostly created in development phase of Stage-Gate model although some demos are built in scoping phase and the technology is fine-tuned in validation phase.

Value created by business plan is another topic that hasn't been addressed properly. Although business plan doesn't create use value as such it is the tool for company level value creation. As innovative companies have higher company value (Cooper 2001 pp. 7-8, Trott 2005 pp. 5) business plan is the tool that can be used to communicate company's future to stakeholders such as analysts and shareholders. This can have significant impact to company valuation.

6.3 Value creation of software process

In section 2.1 was mentioned that software process can be considered as dynamic capability (Mäkelä, Oza & Kontio 2009), thus value creating activity. However, compared to original definition of value creation (please, refer to *section 2.1*), value creation of software process is multifaceted. Main issue is that who is the buying customer and is customer buying technical features or man power to implement their own requirements, product or solution? *Table 18* below summarizes this issue.

Table 18 - Analysis summary for software process value creation

Customer is buying	Customer and supplier roles and expectations	Value creation of software process
Mature technology / product for known problem	Customer has known problem and seeks known solution (mature product). Supplier sells its technology which can be considered whole product including all needed features professional services, support and maintenance.	Software process creates underlying technology for the product but doesn't address whole product issues like services, training and support and maintenance. Software process creates technical product which is only part of use value.
Project (fixed scope, fixed price)	Customer has already created set of requirements (potentially with the help of supplier or 3rd party) which needs to be implemented by a project. Customer is looking supplier that can implement the requirements with best combination of price, schedule and risk.	As use value is already defined at the beginning of the project software process creates only the technical functionality. Software process with strong fixed scope, fixed price project management is the use value.
Team (agile)	Customer problem is somehow defined but requirements are unclear. Customer buys agile team to explore the problem domain and to help customer to learn the correct solution.	Here (agile) software process creates the use value as customer benefits from the outputs of each iteration.

Vendor's solution that breaks customer's status quo	Customer is not originally looking to solve the problem but supplier's solution brings so much use value that customer changes its plans.	Customer buys solution which brings business level benefit. Supplier's software process creates technical features which are only part of the whole product and use value.

As seen in table above software process creates use value only in certain scenarios where customer buys software development. In B2B software product business this is rarely the situation. Although software process is complex task and requires management attention it is not usually such dynamic capability that creates use value as it requires marketing and sales function to drive the use value creation. Therefore earlier proposal (Mäkelä, Oza & Kontio 2009), to consider software process as dynamic capability holds only partially.

6.4 Stage-Gate vs. Chasm model

Stage-Gate model covers the tasks that are included in Trott's definition of innovation (see *section 2.3*) which are idea, development and commercialization. However, Moore's Chasm model explains that biggest challenge in innovation is after first commercialization to transform the value proposition in to whole product which can be effectively sold to mainstream customers. Both, Stage-Gate model and Trott's definition of innovation ends in situation where first early adopter customers have bought the product but product is not entered mainstream market (see *Figure 23*).

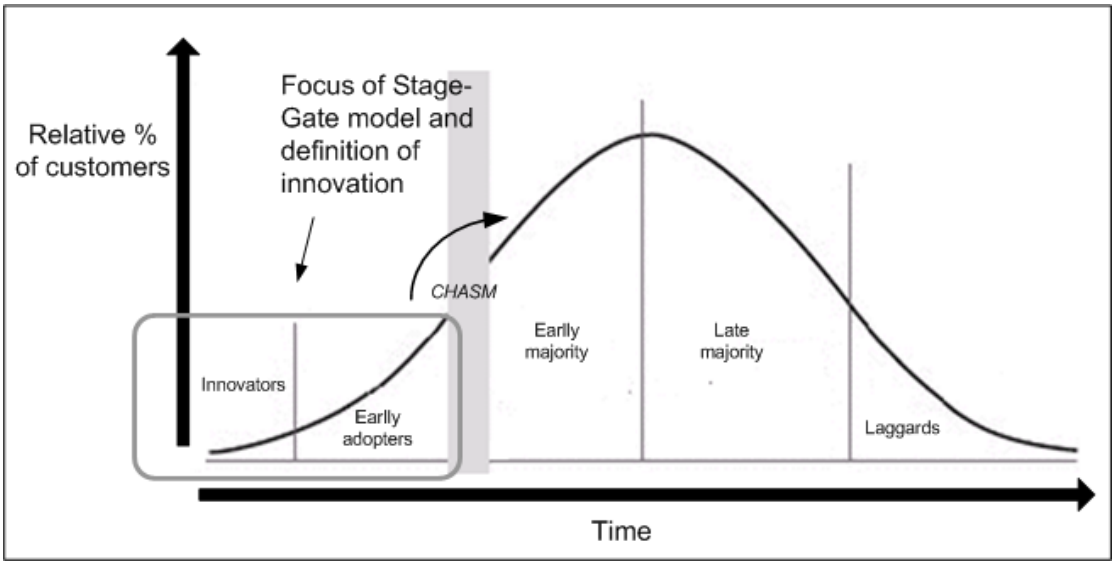

Figure 23 - Focus of Stage-Gate and definition of innovation in Chasm model

Both, Stage-Gate model and Trott's definition of innovation don't give any guidance or tools how to enter to mainstream market. Trott's definition of innovation states that commercialization is target for innovation activity but also failure in the marketplace is option. Both assume that after first commercialization product is ready for mainstream market and there is sales channel that can sell the product efficiently. Also, sales channel development is excluded in both models.

6.5 Stage-Gate vs. Schumpeter and Chasm model

As discussed in previous section (*6.4*) Stage-Gate model and definition innovation address only early market in Chasm model. Schumpeter's Creative Destruction theory can also be linked to Chasm model. Schumpeter defined Creative Destruction:

> *"Companies seek to gain monopoly profits by creating innovations that destroy the current business models and create new ones"*

> Also, *"from company's perspective this means that if competitors have created similar offerings and capabilities, company's competitive advantage has been lost"*

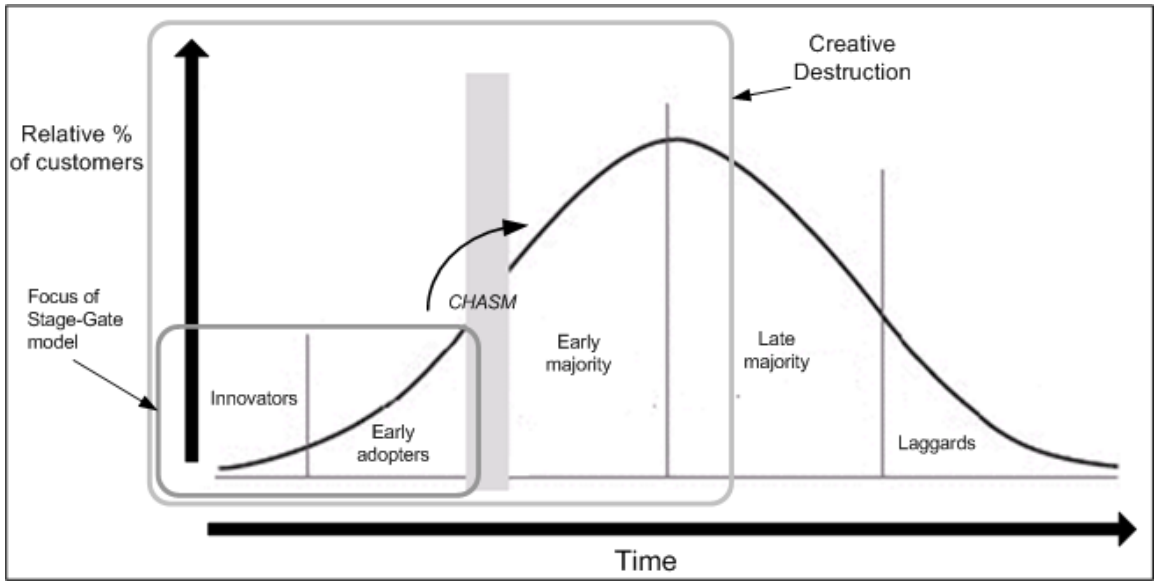

Figure 24 - Stage-Gate, Chasm model and Creative Destruction

Figure 24 above describes how Creative Destruction model fits with Chasm and Stage-Gate models. As Creative Destruction means creating new market and destroying the existing way of doing business in Chasm model it covers the first part of the bell curve. First part of bell curve means new market creation and rapid growth of the company. Late majority and laggards are customers where above average profits are lost and competitive advantage has been normalized partly because of the nature of customers but also because of the competition.

Hockey stick curve, which is many times used as a way to describe sales and profit growth of new venture, can also be fitted to Chasm model. Innovators and early adopters represent the early flat part of hockey stick and early majority represents the aggressive growth of number of customers, revenue and profit.

In addition, Stage-Gate, Chasm model and Creative Destruction can be linked together by looking cost and profit curves. All, Stage-Gate, Chasm and Creative Destruction models define that product development and selling to innovators and early adopters are cost for the company and real profits comes with fast

growth, meaning early adopters. With late majority and laggard customers competition have normalized the profits. See *Figure 25* below.

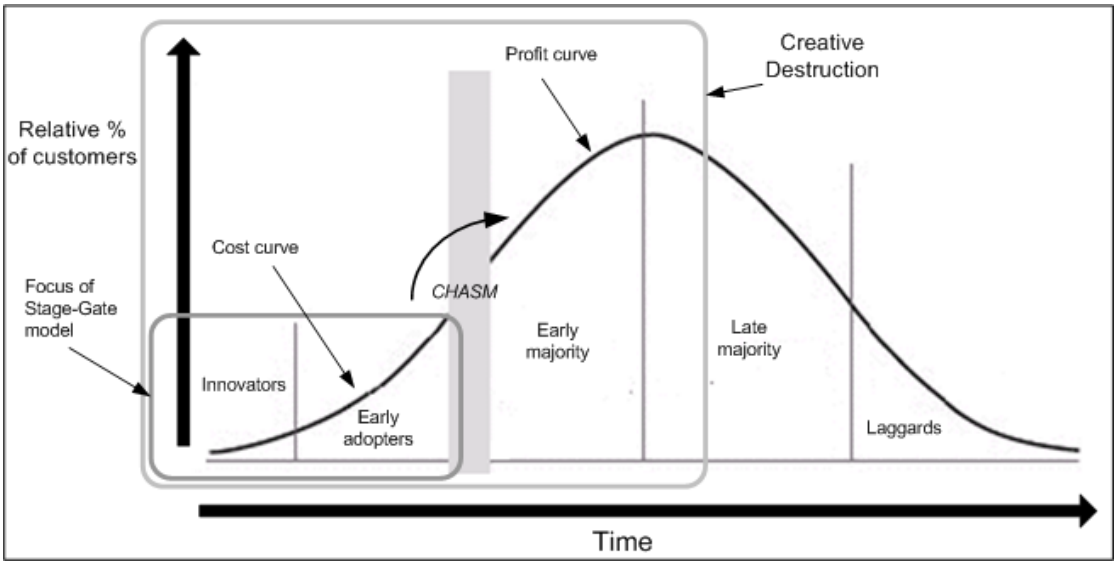

Figure 25 - Cost and profit curve

6.6 Stage-Gate, Chasm model, Creative Destruction with Dominant Design theory

Dominant Design theory (see *section 3.2*) can be linked with Stage-Gate, Chasm and Creative Destruction models. Selection phase of dominant design (i.e. whole product in Chasm model) has to happen prior crossing the chasm. From market perspective variation and era of ferment phases happen with innovator and early adopter customers who are developing and trying different technologies and architectures that companies are selling and developing.

However, technology is not able to reach early majority if it is not dominant design and whole product from customer's point of view. Era of incremental change happens through whole life-cycle from early majority to laggards as they buy dominant design which is being incrementally developed by the companies. *Figure 26* below describes this.

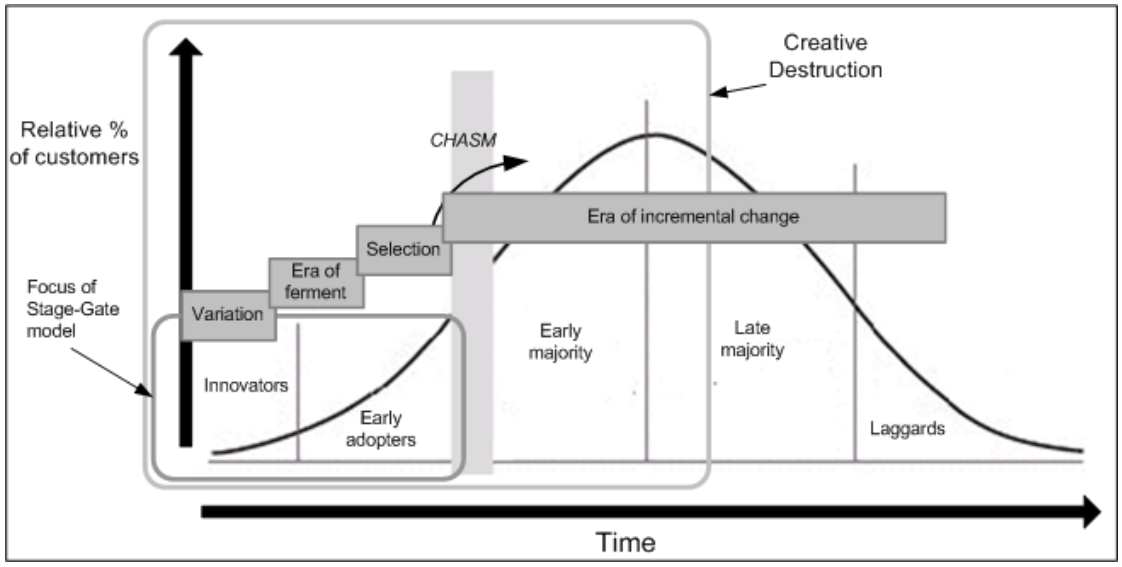

Figure 26 - Dominant Design theory with Stage-Gate, Chasm and Creative Destruction Models

6.7 Sales Channel

Interestingly, most of the innovation and product development literature don't take into account sales channel or the need to develop such. *Table 19* below gives summary how methodologies analyzed in this study deals with sales channel.

Table 19 - Role of sales channel in different models

Model	Role of Sales Channel
Value Creation & Dynamic Capabilities	Sales channel is assumed to exist and being capable of capturing exchange value. Sales channel has narrow and focused role to negotiate and close deals.
Creative Destruction	Sales channel is primary actor as Creative Destruction is the process of capturing exchange value in large scale. Creative Destruction assumes that innovation exists (i.e. use value is created) and it is sold effectively in large numbers and with good profit.
Historical innovation models	Sales channel is part of marketing function and not really in the focus in the models
Paul Trott's definition of innovation	Assumes that sales channel exists and can sell the new product. Failure in marketplace is accepted. Definition doesn't tell that sales channel should be developed and managed properly in order safeguard success in marketplace. Failure in the marketplace happens automatically if sales channel development is neglected.
Stage-Gate	Assumes that sales channel exists and can sell the new product.

B2B sales literature	By definition sales channel is the primary focus. Sales is actively probing, doing market research, giving feedback to R&D and corporate strategy. Sales is active player to creating use value (i.e. value proposition in sales terminology) and also closes the deals. Literature also focuses how sales channel should be organized and how to reach customers in global scale.
Crossing the Chasm	Sales channel is primary actor as the focus of the model is technology maturity and value proposition. Role of sales with innovator and early adopter customers is to sell vision, establish partnership and joint learning. Challenge is to cross the chasm, meaning creating the whole product and mature value proposition. With early majority and rest of the customer groups sales channel's role is to reach effectively all customers and use whole product (mature value proposition) message to customers.

Table above suggests that one of the fundamental issues in value creation and innovation literature is the lack of sales channel development. Current literature takes the view that sales channel development is not needed or the proper sales channel is assumed to exist and it doesn't need any further development. Based on my professional experience I believe that this is untrue. Well-performing sales channel is as fundamental element for the success of new technology as is the technology itself.

Sales channel is also the entity that captures the ROIC value. As there will be no return to invested capital without sales channel it is surprising how little focus sales channel development receives in innovation literature.

6.8 Product Development Waste and Software Process

Cooper (2001 ch. 2) reports that significant amount of R&D efforts do not create value (see also Arthur D Little 2005, *Figure 9*). Moore (2005 ch. 1) explains that investments from R&D efforts results to following (see *Figure 27* below):

- Differentiation, genuine competitive differentiation
- Neutralization, neutralizing competitors' competitive advantage
- Productivity, increasing in-house productivity
- Waste, doesn't produce any benefits

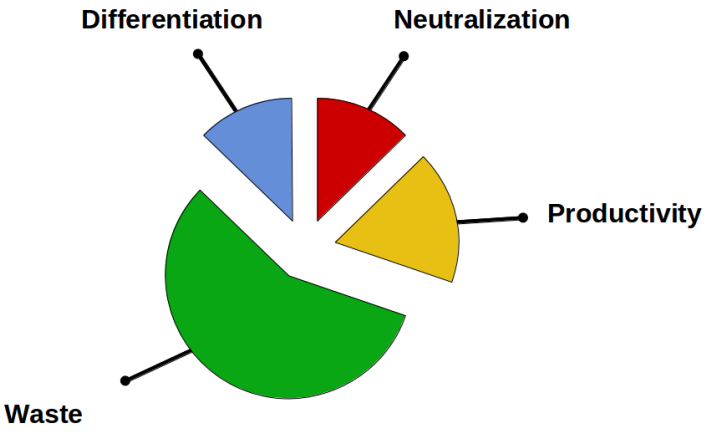

Figure 27 - Outcomes from R&D efforts (Moore 2005 pp.6)

Software engineering literature reports similar findings. De Luca (2003) reports that 64 % of implemented software features in average are never or rarely used and 20 % are used always or often (see *Figure 28* below).

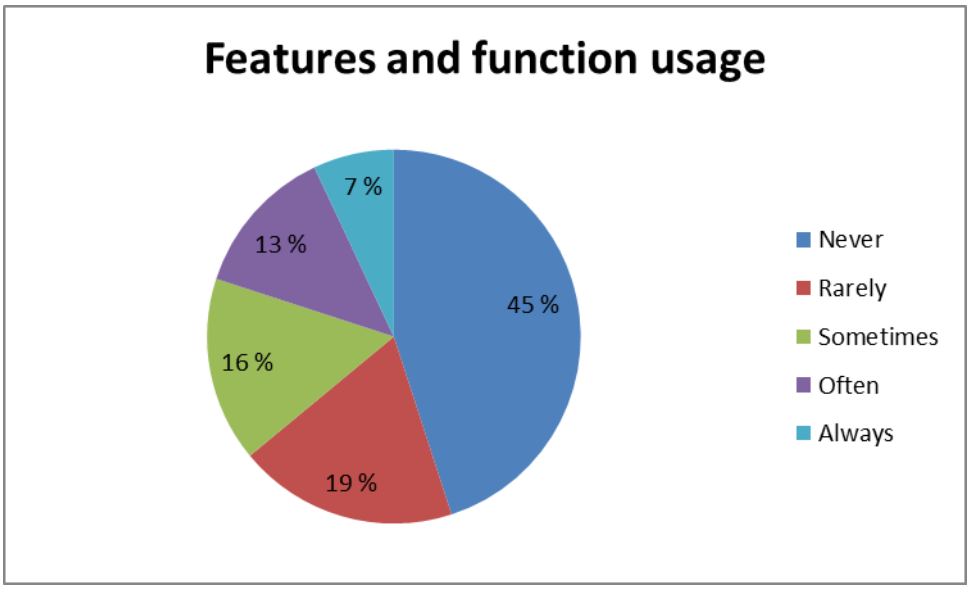

Figure 28 – Actually used features in software (De Luca 2003)

From value creation and value capture point of view this is an issue as majority of R&D efforts do not produce tangible value. As was discussed in *chapter 4*

Business to Business Sales customers are buying business benefits and not technical features. Both software process models say that solving this issue is the task whoever owns the requirements document (waterfall process) or the product owner (agile process). Software process itself can't solve this.

The issue is clearly how to define the right scope for product development projects. From software process point of view the challenges are following:

- In waterfall process the scope is defined in requirements document. The challenge is to know all the needed and value adding features at the point of time when requirements document is written.
- In agile process, especially when using Lean Development Principles (see e.g. Poppendieck 2002), decision what to implement should be done as late as possible in order learn as much as possible to avoid developing features that are not needed. The challenge here is that learning happens in the course of project and cost and schedules are unknown at the time when decisions about the project are being made.

Lean software development eliminates waste by removing total cost, schedule and scope responsibilities from the equation as they do not bring value from software development point-of-view (see *Figure 29* below). However, removing those items doesn't mean that they are unnecessary from business point-of-view. Removing only means that responsibility is moved to other stakeholders.

Waste in software process	Removing waste in agile process
Extra Features	Develop only for today's stories
Requirements	Story cards are detailed only for the current iteration
Extra Steps	Code directly from stories; get verbal clarification directly from customers
Finding Information	Have everyone in the same room; customer included
Defects Not Caught by Tests	Test first; both developer tests and customer tests
Waiting, Including Customers	Deliver in small increments
Handoffs	Developers work directly with customers

Figure 29 – Waste in software process and ways to remove it (adapted from Poppendieck 2002)

Cooper (1999 ch 1) also stated that companies tend to invest too much in product maintenance compared to developing differentiation and competitive advantage. The issue here is that different stakeholders of the software product have different needs. Many of the software features are just needed in order to make the software operational. Thus those features are necessity for end users and operational personnel, even if used extremely rarely. However, they not bring any additional value in sales situations and calculating their value is impossible in practice. In order avoid spending development effort to assumed but at the end unnecessary features, learning during the project is needed and therefore waterfall process doesn't apply as fixed scope - fixed price project is not possible.

7. Conclusions

This study examined software product company's value creation and value capture process from R&D and sales perspective. Goal was to analyze innovation management and new product development models, B2B sales practices and software development process models and how they fit with value creation and value capture theory. This chapter summarizes and explains the findings, identifies theoretical gaps, introduces practical managerial implications and proposes topics for further research.

Study made several findings. Firstly, the definition of innovation and Stage-Gate model are not taking all necessary phases in to account to launch new software product as they exclude sales channel development and they don't understand dominant design theory and whole product concept (see *section 7.1* below).

Secondly, used software development process creates major implications to practical new product development and commercialization activities as traditional and agile software processes treat risk, learning and long term planning differently (see *section 7.2*).

Thirdly, the concept of value creation and dynamic capability remains problematic and there are multiple issues (see *section 7.3* below).

- The concept of use value is vague and how that relates to value capture is problematic.
- In practice there is lot of waste in innovation process as not all activities are really value creating.
- Practical managerial processes used by companies and dynamic capability concept do not fit very well together.

7.1 Phases Missing in Innovation Process

Both, definition of innovation and Stage-Gate model exclude sales channel development phase and the necessary technology evolution towards mature

technology in order to break in to mainstream markets (to enable Creative Destruction). Both models assume that proper sales channel exist and it can efficiently sell new product. Also, both models assume that when product is launched it is mature enough for sales channel and have reached dominant design status.

Lack of sales channel development and not understanding necessary technology evolution for whole product causes significant managerial problems. Making assumption that company's sales channel can sell new product without considerable training and without support will lead to significant business performance problems. It might be that company has spent several years of developing new product with its first customers. It is impossible for sales channel to develop similar understanding of product's value proposition and target customers and their processes in fraction of time compared to the time spent by R&D.

Also, if sales channel receives immature product from R&D sales channel has to deal with technology evolution and dominant design issues. In practice this means that sales channel has to take responsibility to create whole product and carry lot of value proposition and technical architecture related development responsibilities in addition to their normal work (which is selling as efficiently as possible and managing customer relationship). For global sales channel with hundreds or thousands people that do not operate jointly together, fixing value proposition and dominant design issues are just impossible. Sales channel can only sell effectively products that are mature, their value propositions are trial and tested and they represent dominant design and whole product that mainstream customers can buy.

7.2 Implications of Used Software Process

Used software process cause significant implications how new product development and innovation process should be managed. Main issue is that traditional waterfall process assumes known problem – known solution project that can be ordered and delivered with fixed price, schedule and cost. Agile

process assumes exploratory project where learning for both project team and product owner is essential.

From investment decision perspective waterfall process is nice as it commits to deliver agreed functionality with agreed schedule and cost. In waterfall process the party who implements the software prices the schedule and cost risk in the price. Pure agile process will not take any schedule and cost risk and transfers that to product owner. Another issue with agile process is that it lacks long term planning principle to support investment decisions.

However, the main problem is that new product development is exploratory activity by nature and learning has to be done in one way or another. Waterfall process assumes that learning is already done prior entering to software development phase. Agile process assumes that learning will be done during the software development project.

Value creation and value capture perspective learning and product development phase should be done as fast as possible in order to reach mature product state for efficient sales and profits. Agile process optimizes learning and time-to-market but doesn't give help for long term planning and investment decisions. So there has to be planning phase before development starts to understand main elements of technology, value proposition, business potential and competition (as defined in Stage-Gate model). Also, on-going planning and go/no-go control mechanism are needed to steer new product development process to right direction.

7.3 Vagueness of Value Creation and Dynamic Capability Theory

Originally value creation and value capture theory had value chain focus i.e. how value was created by different companies in value chain through the series of commercial exchanges. Later the theory evolved to different levels of value creation of the company. This study focused to value creation of R&D and sales and marketing so the value creation and value capture is two folded; use value for customer and exchange value for supplier, and value created for company's

shareholder by the increase of sales and profits and by the increase of share price due to better future business outlook.

As there are lot of software industry and B2B sales specific processes and practices the value creation and value capture theory explains only on very high level how B2B software product companies should manage their innovation processes. The concept of use value is problematic. Although, it describes customer's understanding the practical business value of vendor's solution it doesn't tell anything how to develop and market that technology. Implicitly, the concept of use value assumes that customer knows what they are looking for, understands the vendor's solution correctly and can forecast solution's benefits correctly for commercial negotiations. However, from new product development perspective the majority of the challenges are happening before reaching to state where dialogue about use value between supplier and customer can be done in a way that the value creation and value capture model suggests.

Not all processes and activities used by companies are really creating value as there is lot of waste in all areas of innovation activity. If we exclude quality issues of actual technical software development, major source for problems is poor decision making around value proposition (i.e. features, priorities, technology evolution towards dominant design) and how to sell it (i.e. technology maturity and sales channel capability).

Also, both academic and management literature say that value is created in various levels by various actors, such as project team when implementing software. However, isolating different actors in value creating process blurs the overall picture and it creates assumption that all activities are always value creating.

In addition, dynamic capability theory remains to be weak compared to practical managerial processes used by software product companies. It looks like that dynamic capability is more high-level term that covers both strategic decision making (including planning activity) and practical execution of chosen strategic

projects. Dynamic capability doesn’t give any practical guidelines for companies how to do strategic decision making and how to implement chosen projects.

To conclude, all practical and managerial processes used by software product companies are industry specific with their own history and research traditions. Therefore, it is fair to say that value creation and value capture and dynamic capability theory can’t and shouldn’t overcome those.

7.4 Limitations of the Study

This study has its limitations. Firstly the methods and principles analyzed in this study are chosen by author using personal expertise, experience and judgment. Selection of the analyzed methods is not based on any research made among the industry practitioners so there might be methods and principles which aren’t covered but are relevant in the context. Also, some covered principles and methods might not be used widely in the industry at all. However, author believes that selected methods and principles represent what should be the industry best practice thus not all of them are in wide use.

Secondly, the nature of the study is desk research, meaning that the analysis is based on literature review and author’s own expertise and judgment. As there are no qualitative or quantitative research created as an input to this study, the viability of the results rely solely on author’s analysis.

Thirdly, the study examines innovation process in a way that hasn’t been done earlier. This means that there is no established research tradition that combines new product development process, B2B sales and software development together. As the scope of the study is extensive further and stronger analysis of certain findings might be needed. Also it might be possible that there are some relevant findings that remain uncovered.

7.5 Ideas for Future Research

This study concluded that value creation and value capture and dynamic capability theories are weak when compared to practical processes used by

companies. Also the results of this study are similar with Bowman & Ambrosini's (2009) critique that dynamic capability theory is tautological (see *section 2.2*). This suggests that future research should be guided to understand the usefulness of the whole value creation and dynamic capability theory.

The other idea for future research is the time aspect. As innovation is financially betting money for better return compared to alternatives with certain risk, time aspect of whole innovation process becomes crucial. However, there isn't research made how development project, sales cycle and delivery project length affect the innovation process and return on investment. For example, just 6 months increase to sales cycle has tremendous effect as feedback loop to R&D and strategic decision making gets delayed and the sales channel is using that time for creating costs and not bringing money to the company. This is especially true in cases where product is not reached to dominant design status and where learning responsibility is transferred to sales channel.

Third idea for future research is to understand the motivation and goals to do innovation of each of actors involved. This means the motivation of top-management, R&D, marketing and sales functions in the innovation process might not be automatically aligned. For example, it might be that top-management is mainly looking to increase shareholder value. This could mean that good looking business plans and external communication creates most of the value as they know that bringing new product successfully to the mainstream market takes so long that they are not probably working with the company anymore. On the other hand, R&D might be very reluctant to take any risk and commit to any schedule and cost as they know that developing new technology is difficult. Instead of, R&D personnel might be focusing more to learning new skills and use them to get better positions in other companies thus they want to try new things. Also sales channel might be more looking for products that are mature and easy to sell in order to reach their sales quotas so they might be very reluctant learn new and immature technologies.

8. References

Abrahamsson, Salo, Ronkainen & Warsta (2002), Agile Software Development Methods – Review and Analysis, VTT Publications 278, Espoo 2002

Agile Alliance 2001, www.agilemanifesto.org, retrieved 26th March 2011

Ambysoft (2008), Software Development Project Success Survey 2008, Ambysoft Corp. http://www.ambysoft.com/surveys/success2008.html, retrieved 14th April 2011

Ambysoft (2010), IT Project Success Survey 2010, Ambysoft Corp. http://www.ambysoft.com/surveys/success2010.html, retrieved 14th April 2011

Anderson, Narus & Rossum (2006), Customer Value Propositions in Business Markets, Harvard Business Review, March 2006

Anderson & Tushman (1990), Technological Discontinuities and Dominant Designs: A Cyclical Model of Technological Change, Administrative Science Quarterly, Vol. 35, No. 4. (Dec., 1990), pp. 604-633.

Arnold (2004), Policymaking - Evaluating research and innovation policy: a systems world needs systems evaluations, Research Evaluation, volume 13, number 1, April 2004, pages 3–17

Arnold & Bell (2001), Some New Ideas about Research for Development, Partnership at the Leading Edge: A Danish Vision for Knowledge, Research and Development, Danish Ministry of Foreign Affairs, p.278-316 (2001)

Arthur D Little (2005), Global Innovation Excellence Survey 2005, www.adl.com, retrieved 17th May 2011

Asher (2009), How To Win a Pitch, Persuasive People Press

Atkinson (1999), Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria, International Journal of Project Management Vol. 17, No. 6, pp. 337-342, 1999

Atwood (2004), Gold Plating, http://www.codinghorror.com/blog/2004/12/gold-plating.html, retrieved 7th June 2011

Aoyama (2001), Continuous and Discontinuous Software Evolution: Aspects of Software Evolution across Multiple Product Lines, Proceedings of the 4th International Workshop on Principles of Software Evolution, ACM

Benington (1983). Production of Large Computer Programs, IEEE Annals of the History of Computing (IEEE Educational Activities Department), 1 October 1983, 350–361

Boehm & Turner (2003), Balancing Agility and Discipline: A Guide for the Perplexed, Pearson Professional Educational

Bowman & Ambrosini (2000), Value Creation versus Value Capture: Towards a Coherent Definition of Value in Strategy, British Journal of Management, Vol. 11, 1–15 (2000)

Bowman & Ambrosini (2007), Firm value creation and levels of strategy, Management Decision Vol. 45 No. 3, 2007 pp. 360-371

Bowman & Ambrosini (2009), What are dynamic capabilities and are they a useful construct in strategic management? International Journal of Management Reviews (2009) doi: 10.1111/j.1468-2370.2008.00251.x

Brown & Eisenhardt (1997), The Art of Continuous Change: Linking Complexity Theory and Time-paced Evolution in Relentlessly Shifting Organizations, Administrative Science Quarterly, 33 March 1997

Chesborough (2003a), The Era of Open Innovation, MIT Sloan Management Review, Spring 2003, issue 44 vol 3

Chesborough (2003b), Open Innovation: The New Imperative for Creating and Profiting from Technology, Harvard Business School Publishing, 2003

Cheverton (2006), Global Account Management, Kogan Page

Cohen & Levinthal (1989), Innovation and Learning: The Two Faces of R&D, The Economic Journal, Vol. 99, No. 397 (Sep., 1989), pp. 569-596

Collins (2001), Good To Great: Why Some Companies Make the Leap... and Others Don't, Harper Collins, New York

Cooper (1979), The Dimensions of Industrial New Product Success and Failure, Journal of Marketing, Vol 43 (Summer 1979), 92-103

Cooper (1980), Project NewProd: Factors in New Product Success, European Journal of Marketing, Volume 14 issue 5/6, pp. 277 – 292

Cooper (1985), Selecting Winning New Product Projects, Journal of Product Innovation Management, 1985: 2. 34-44

Cooper (1994), New Products: The Factors that Drive Success, International Marketing Review, 1994; 11, 1

Cooper (1999), Product Leadership: Creating and Launching Superior New Products, Perseus Books Cambridge Massachusetts

Cooper (2001), Winning at New Products: Accelerating the Process from Idea to Launch, 3rd Ed., Basic Books New York

Cooper & Edgett (2008). Maximizing Productivity and Product Innovation, Industrial Research Institute Inc, March-April 2008

Cooper, Edgett & Kleinschmidt (2001), Portfolio Management for New Product Development: Results of an Industry Practices Study, R&D Management, 31, 4, 2001

Cova, Ghauri & Salle (2002), Project Marketing – Beyond Competitive Bidding, Wiley

Cutting (2007), Scope Creep, http://www.pmhut.com/?s=%22Scope+Creep+Part%22, retrieved 7th June 2011

D'Aveni (1994), Hypercompetition: Managing the Dynamics of Strategic Maneuvering. Free Press: New York.

D'Aveni (1995), Coping with hypercompetition; Utilizing the new 7S's framework, Academy of Management Executive, 1995 Vol. 9 No. 3

D'Aveni, Dagnino & Smith (2010), The Age of Temporary Advantage, Strategic Management Journal. 31: 1371–1385 (2010)

Deeds, DeCarolis & Coombs (2000), Dynamic Capabilities and New Product Development in High Technology Ventures and Empirical Analysis of New Biotechnology, Journal of Business Venturing 15, 211–229

De Luca (2003), Requirements – The Budgeting Syndrome, http://www.featuredrivendevelopment.com/node/614, retrieved 9th June 2011

Eden & Mens (2006), Measuring Software Flexibility, Measuring Software Flexibility, IEE Software, Vol. 153, No. 3 (Jun. 2006), pp. 113–126. London, UK: The Institution of Engineering and Technology.

Eclipse Foundation (2010), The Open Source Developer Report: 2010 Eclipse Community Survey, June 2010

Effective Agile (2009), Scrum Implementation Workshop, http://effectiveagiledev.com/AgileTraining/ScrumImplementationWorkshop/tabid/74/Default.aspx, retrieved 7th June 2011

Eisenhardt & Martin (2000), Dynamic Capabilities: What Are They?, Strategic Management Journal, 21: 1105–1121 (2000)

Fichman & Moses (1999), Incremental Process for Software Implementation, Sloan Management Review 1999, Vol. 40, Issue 2.

Forss (2009), Confessions of Serial Product Owner, annaforss.spaces.live.com, retrieved 3rd November 2010

Galen (2009), Scrum Product Ownership: Balancing Value from Inside Out, RGCG, LLC

Helfat, Finkelstein,Mitchell, Peteraf, Singh, Teece & Winter (2007). Dynamic Capabilities: Understanding Strategic Change in Organizations. London: Blackwell.

Higsmith (2010), Agile Project Management – Creating Innovative Products, Addison-Wesley, 2nd Edition 2010

Highsmith (2010b), Beyond Scope, Schedule and Cost – The Agile Triangle, http://www.jimhighsmith.com/2010/11/14/beyond-scope-schedule-and-cost-the-agile-triangle/, retrieved 7th June 2011

IBM (2011), IBM Rational Unified Process, http://www-01.ibm.com/software/awdtools/rup/#, retrieved 7th June 2011

Kim & Mauborgne (2005), Blue Ocean Strategy: How to Create Uncontested Market Space and Make the Competition Irrelevant, Harvard Business School Publishing

Konrath (2006), Selling to Big Companies, Kaplan Publishing

Kroll (2005), Key Principles for Business-Driven Development, http://www.ibm.com/developerworks/rational/library/oct05/kroll/, retrieved 7th June 2011

Lehmann (2006), The Creeping Scope, http://www.visionarytools.com/decision-making/incomplete-contracts-scope-creep.htm, retrieved 7th June 2011

Lehman & Belady (1985), Program Evolution: Process of Software Change, Academic Press Professional, San Diego 1985

Lepak, Smith & Taylor (2007), Value Creation and Value Capture: A Multilevel Perspective, Academy of Management Review 2007, Vol. 32, No. 1, 180–194

Lee, Venkatraman, Tanriverdi & Iyer (2010), Complementarity-Based Hypercompetition in the Software Industry: Theory and Empirical Test 1990-2002, Strategic Management Journal, 31: 1431–1456 (2010)

Marsh & Stock (2003), Building Dynamic Capabilities in New Product Development through Intertemporal Integration, Journal of Product Innovation Management 2003;20:136–148

Messerschmitt & Szyperski (2003), Software Ecosystem – Understanding An Indispensable Technology and Industry, MIT Press, 2003

Moore (1999), Crossing the Chasm: Marketing and Selling Disruptive Products to Mainstream Customers, HarperCollins Publishing

Moore (2005), Dealing With Darwin: How Great Companies Innovate at Every Phase of Their Evolution, Portfolio

Mountaingoat Software (2011), Scrum Presentations, http://www.mountaingoatsoftware.com/presentations-scrum?page=1, retrieved 26th March 2011

Mäkelä, Oza & Kontio (2009), Dynamic capabilities in the software process, International Journal of Information Technology and Management, Vol. 8, No. 1

Möller (2006), Role of competences in creating customer value: A value-creation logic approach, Industrial Marketing Management 35 (2006) 913–924

O'Cass & Ngo (2009), Examining the Firm's Value Creation Process: A Managerial Perspective of the Firm's Value Offering Strategy and

Performance, British Journal of Management, 22: no. doi: 10.1111/j.1467-8551.2010.00694.x

Page (2002), Hope is Not a Strategy, Nautilus Press

Payne & Holt (2001), Diagnosing Customer Value: Integrating the Value Process and Relationship Marketing, British Journal of Management, Vol. 12, 159–182 (2001)

Poppendieck (2002), Principles of Lean Thinking, http://www.leanessays.com/2002/11/principles-of-lean-thinking.html, retrieved 9th June 2011

Porter (1979), How competitive forces shape strategy, Harvard Business Review, March-April 1979

Porter (1987), From competitive advantage to corporate strategy, Harvard Business Review, Vol. 65 No. 3, pp. 43-60.

Product Development Institute (2011), http://www.prod-dev.com/stage-gate.php, retrieved June 7th 2011

Richardson (1998), Stop Telling, Start Selling – How to Use Customer-Focused Dialogue to Close Sales, McGraw-Hill

Rogers (1962), Diffusion of Innovations, Free Press, New York

Rosenkopf & Tushman (1994), The Coevolution of Community Networks and Technology: Lessons from the Flight Simulation Industry, Industrial and Corporate Change, 7 pp. 311-346.

Royce (1970), Managing the Development of Large Software Systems, Proceedings of IEEE WESCON 26 (August): 1–9.

Schamansee (2000), Antitrust Issues in Schumpeterian Industries, The American Economic Review, Vol. 90, No. 2, Papers and Proceedings of the 102th Annual Meeting of the American Economic Association (May, 2000), pp. 192-196

Schiffman (2002), 25 Sales Skills They Don’t Teach in Business School, Adams Media

Schiffman (2004a), Closing Techniques (That Really Work!), Adams Media

Schiffman (2004b), 25 Most Dangerous Sales Myths, Adams Media

Schiffman (2007), Sales Presentation Techniques (That Really Work!), Adams Media

Schumpeter (1950), Capitalism, Socialism, and Democracy, Harper & Row (New York)

Schwaber & Beedle (2002), Agile Software Development with Scrum, Pearson Education Limited

Shapiro & Varian (1999), Information Rules: A Strategic Guide to the Network Economy. Harvard Business School Press: Boston

Sirmon, Hitt & Ireland (2007), Managing Firm Resources in Dynamic Environments to Create Value: Looking Inside the Black Box, Academy of Management Review 2007, Vol. 32, No. 1, 273–292.

Smart (2007), The Waterfall Development Methodology, http://www.mikesmart.com/application_development/waterfall_method.htm, retrieved June 7th 2011.

Spencer & Kirchhoff (2006), Schumpeter and New Technology Based Firms: Towards a Framework for How NTBFs Cause Creative Destruction, International Entrepreneurship and Management Journal, 2006, Vol 2; Number 2, pages 145-156

Teece, Pisano & Shuen (1997), Dynamic capabilities and strategic management. Strategic Management Journal, 18, 509–533.

Trott (2005), Innovation Management and New Product Development, Pearson Education 2005

Trout (2008), Differentiate or Die: Survival in Our Era of Killer Competition, Wiley

Turner (1999), Handbook of Project-Based Management, McGraw-Hill

Tushman & Smith (2002), Innovation Streams, Organization Designs and Organization Evolution, chapter 1 in Tushman & Anderson 2004, Managing Strategic Innovation and Change, Oxford University Press

Wang & Ahmed (2007). Dynamic Capabilities: a Review and Research Agenda. International Journal of Management Reviews, 9, 31–51.

Winter (2003). Understanding Dynamic Capabilities., Strategic Management Journal, 24, 991–995.

Zahra, Sapienza & Davidsson (2006), Entrepreneurship and Dynamic Capabilities: a Review, Model and Research Agenda. Journal of Management Studies, 43, 917–955.

Zollo & Winter (2002), Deliberate Learning and the Evolution of Dynamic Capabilities, Organization Science Vol. 13, No. 3, May-June 2002, pp. 339-351