1 Software Project Management: Setting the Context

Günther Ruhe and Claes Wohlin

Abstract: This chapter is designed as the introduction to the book. It provides the motivation for studying software project management as a response to the increasing variety of software development methodologies. The chapter characterizes software projects and presents ten knowledge areas in software project management. This body of knowledge is described in the Software Edition of the Project Management Body of Knowledge (PMBOK). The chapters of the book are classified in terms of their contribution to these knowledge areas.

The chapter also discusses the multi-disciplinary nature of the project management discipline. Based on some predicted trends for the future of software engineering, a prediction on the future of software project management is given. Finally, an overview of the content and structure of the whole book is presented.

1.1 Motivation

The world is continuously changing. Software and software-intensive systems are among the key drivers of this trend. The speed and magnitude of all these changes is breathtaking. What would happen today if any of the existing telecommunication, health care, financial or logistic systems are not performing securely, safely and reliably? The rapid growth in technology in combination with the strong dependence of products and services from software raises the demand on managing the development and evolution of such systems.

Project management is one of the youngest, most vibrant, and most dynamic fields among different management disciplines. According to the Project Management Body of Knowledge (PMBOK), project management is the "application of knowledge, skills, tools and techniques to project activities to meet the project requirements" (PMI 2013a). Project management is accomplished through the application and integration of 47 logically grouped project management processes divided into five process groups called initiating, planning, executing, monitoring and controlling, and closing.

Software is a direct product of the cognitive processes of individuals engaged in innovative teamwork. Many of the procedures and techniques used in software project management are designed to facilitate communication and coordination among team members engaged in an intellectually intensive work. Software development is often characterized as a learning process in which knowledge is gained and information generated during the project. Dealing with people, conflicts, team building, knowledge sharing, and communication will be determinants of good project management.

Software project management deals with software projects and the challenges of human-based development (as opposed to the more deterministic processes in traditional projects). The higher flexibility in software development approaches puts new demands on the capabilities of software project management. Weaknesses in planning, organizing, staffing, directing and controlling are hard to be counter-balanced by more efficiency in technical development work. As Fred Brooks stated in 1987, "… today's major problems with software development are not technical problems, but management problems" (Brooks 1987).

The principal nature of the challenges in software project management has not changed dramatically in the last 25 years. However, software-intensive systems of the 21st century increasingly vary in their content, size, complexity and their degree of interaction with other systems. The technological and communication infrastructure to develop these systems is hard to compare with the ones available in the past. As a consequence, the concrete content of the project management challenges looks different from 25 years ago.

Beginning from the 1970s and 1980s, traditional plan-driven software development has been replaced and complemented by more adaptive and dynamic approaches. Global (or distributed) software development, open source development, and the application of the different variants of adaptive development techniques have proven successful under various circumstances. The Internet has dramatically enhanced the ability of individuals, teams and organizations to manage projects across continents and cultures in real time (Kwak and Anbari 2008). New paradigms (such as inner source project management) or emerging techniques (such as social media collaboration) provide new opportunities for conducting software project management more successfully than before.

1.2 Characteristics of Software Projects and why Software Project Management is Difficult

Software development is both human-intensive and knowledge-intensive, which makes people the most important asset in any software development endeavour. *Software projects* differentiate from other projects in a number of ways. Consequently, management of software projects cannot be done in the same way as in traditional project management and needs to be adjusted correspondingly. Following (PMI 2013b), some of the main differentiating factors are:

- Software is an intangible product.
- Software is a cognitive and human-based development process that requires sharing of documents.
- Higher degree of uncertainty in the project and product scope.
- Communication and coordination of within software teams and with project stakeholders often lacks clarity.

- Intellectual capital of software personal is the primary asset for software projects and organizations.
- Degree of change of requirements in the course of the software project.
- Creation of software requires innovative problem solving to create unique solutions.
- Initial planning and estimation of software projects is challenging because these activities depend on requirements which are often imprecise or based on lacking information.
- Development and evolution of software-intensive system is challenging because of the high complexity of software based on the enormous number of logical paths in program modules and all the combinations of interface details.
- Exhaustive testing of software is impractical because of the time and related complexity constraints.
- Software development often involves interactions to different vendor products and interfaces to other software.
- Software security is a large and growing challenge.
- Objective measurement and quantification of software quality is difficult.
- Learning and knowledge creation in software development is more difficult because processes, methods and tools are constantly evolving.
- Execution of software is platform dependent and often is an element of a system consisting of diverse hardware, other software, and manual procedures.

Software project life-cycles are models of how software projects pass through phases of development, from its initiation to its closure. The software extension of the PMBOK describes the continuum of software project life cycles ranging from highly predictive to highly adaptive (PMI 2013b). The variation between them is described by the degree of change in requirements (from being specified during initiation and planning to elaboration at frequent intervals during development), control of cost and risk, and involvement of key stakeholders (from involvement at scheduled milestones to continuous involvement).

Practically, all modern development approaches are iterative and incremental (Larman and Basili 2003) in their key nature. The notion of iteration refers to a phase within the development process, while the term of an increment describes a certain stage of the product evolution. An example of an incremental software product development process is given in Figure 1.1.

Kruchten (2011) has proposed a *conceptual model of software development*. The main pillars of that model are four core entities called *Intent*, *Product, Work*, and *People*, which are common entities across all software development projects. In brief, the entities give answer to the following questions:

Intent:	What is the project trying to achieve?
Product:	What is the outcome of the project?
Work:	How to build the targeted product?
People:	Who will be available to perform the work?

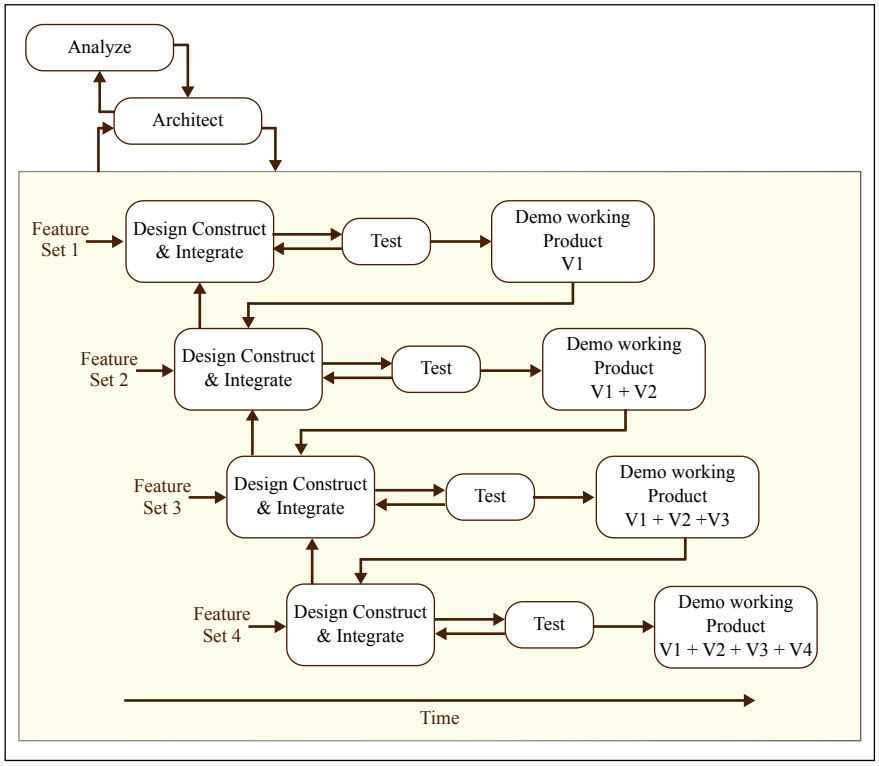

Figure 1.1 Incremental software product development (PMI 2013a)

Each of these four core entities has three attributes: Time, Quality and Risk. The *Time* attribute refers to a description of the execution of the project over time. *Quality* is applied as well to all the four entities. For example, quality of the people refers to the competence and diligence and dedication of the staff assigned to the project. The *Risk* attribute describes the inherent uncertainty of all the entities.

In addition, there is a *Value* attribute for Intent and Product, as well as a *Cost* attribute for People and Work. The majority of cost is associated with the cost of people performing the work. The intended value can be defined in terms of functional and quality requirements. The resulting value can be expressed in financial terms by the Net Present Value of the product.

Besides the common core concepts, there are factors used to differentiate between projects. All these factors can be grouped into *organization-level* and *project-level project factors*:

i. Organization-level

- **a. Business domain:** For what domain the software (-based) product is developed?
- **b. Number of instances:** How many instances of the software (-based) product will be deployed?
- **c. Maturity of organization:** How mature are the processes of the software developing organization?
- **d. Level of innovation:** How innovative is the organization?
- **e. Culture:** In which culture are the projects developed?

ii. Project-level

- **a. Size:** How big is the system under development?
- **b. Stable architecture:** Is a stable architecture in place?
- **c. Business model:** Under which business model the software (-based) product is developed?
- **d. Team distribution:** How many teams and in which configuration are working in the project?
- **e. Rate of change:** How stable is your business environment and how much risks and uncertainties are you facing?
- **f. Age of system:** Greenfield (from scratch) versus brownfield (evolving) software system development
- **g. Criticality:** How many people's safety will be threatened if the system fails?
- **h. Governance:** Who manages the project managers? How much governance is applied to the project?

This classification can serve as guidance to approach the question: Which software project management approach has proven successful and is recommended for which given configuration of project-level and organization-level factors? We do not expect having a single answer in most cases. We also expect having clusters of configurations where certain approaches are recommended. As the pathway to accumulate the knowledge needed to make these decisions, the *empirical paradigm* suggests the replicated application of the making observations, modeling of the real-world phenomena, measuring and analyzing, validation of hypotheses and defining another cycle with slightly changed parameters to confirm existing and creating new knowledge (Basili et al. 1999).

1.3 Ten Knowledge Areas of Software Project Management

The Project Management Body of Knowledge (PMBOK) (PMI 2013a) as a set of standard terminology and guidelines for project management is familiar for most professionals and researchers. A PMBOK knowledge area contains the processes

that need to be accomplished within its discipline in order to achieve an effective project management program. The Software Extension of the PMBOK (PMI 2013b) is based on the PMBOK, but provides an extension towards commonly accepted practices for managing software projects. Software project management processes are grouped into *Initiating, Planning, Executing, Monitoring and Controlling, and Closing*.

The whole body of knowledge of software project management is described by ten knowledge areas. Each area is defined by a set of associated processes. In what follows, and aligned with the Software Extension of the PMBOK (PMI 2013b), we describe the main content of these knowledge areas. For five areas being closest to the content of the book, a more detailed description including inputs, outputs, and the main tools and techniques (to be) used for the respective processes is given. These knowledge areas are:

- Time management
- Cost management
- Human resource management
- Communications management
- Risk management

1.3.1 Integration Management

This area coordinates other areas to work together throughout the project and includes the processes and activities to identify, define, combine, unify and coordinate the various processes and project management activities within the project management process groups. This knowledge area known as key element that unify, consolidate and articulate all project components the processes in this area crucial to controlled project execution through completion, successfully managing stakeholder expectations, and meeting requirements. Project integration management includes making choices about resource allocation, making trade-offs among competing objectives and alternatives, and managing the interdependencies among the project management Knowledge Areas. This knowledge area emphasizes the generally accepted role of a project manager: Performing coordination and bringing all the pieces (the deliverables of the project) together. It also refers to integration of processes and activities (PMI 2013b).

1.3.2 Scope Management

A set of processes used to ensure that the project includes all of the requirements and no new requirements are added in a way that could harm the project this knowledge area includes the processes required to ensure that the project includes all the work required, and only the work required, to complete the project successfully and includes setting clearly defined project objectives, defining major project deliverables, and controlling changes to those deliverables. Managing the scope is primarily concerned with defining and controlling what is and is not included in the project.

For software, the definition of product scope includes features and quality attributes that are needed and desired by stakeholders. Product scope can be used to estimate project scope and constraints on project scope may determine product scope. Constraints on both of these two may require trade-off among features, quality attributes, schedule, budget, resources and technology.

1.3.3 Time Management

Processes required to manage the timely completion of the project and to ensure that the project is completed on schedule. In software projects this knowledge area is driven by risk, resource availability, business value and the scheduling methods used. Specific scheduling methods for software projects introduced by (PMI 2013b) including structured scheduling, schedule as independent variable, iterative scheduling with a backlog, on-demand scheduling, portfolio management scheduling. All the respective processes with their inputs, outputs, and tools and techniques (to be) used are summarized in Table 1.1.

Table 1.1 Overview of project time management processes (PMI 2013b)

Process Name	Inputs	Outputs	Tools and Techniques
Plan schedule management	– Project management plan – Project charter – Enterprise environmental factors – Organizational process assets – Safety and security issues	– Schedule management plan	– Expert judgment – Analytical techniques – Meetings
Define activities	– Schedule management plan – Scope baseline – Enterprise environmental factors – Organizational process assets – Additional factors	– Activity list – Activity attributes – Milestone list	– Decomposition – Rolling wave planning – Experts judgment – Story breakdown structure – Storyboards – Use cases
Sequence activities	– Schedule management plan – Activity list – Activity attributes – Milestone list – Project Scope statement – Enterprise environmental factors – Organizational process assets – Architectural and IV & V constraints – Safety and security analyses	– Project schedule network diagrams – Project document updates – Feature sets – Release plans – Architectural and nonfunctional dependencies	– Precedence diagramming method (PDM) – Dependency determination – Applying Leads And Lags – SAIV and time boxing – Work in progress limits and classes of service – Feature set evaluation – Service level agreements
Estimate activity resources	– Schedule management plan – Activity list – Activity attributes – Risk register – Activity cost estimates – Resource calendars – Enterprise environmental Factors – Organizational process assets	– Activity resource requirements – Resource breakdown structure – Project document updates	– Experts judgment – Alternatives analysis – Published estimating data – Bottom-up estimating – Project management software

Table 1.1 (continued)

Process Name	Inputs	Outputs	Tools and Techniques
Estimate activity durations	– Schedule management plan – Activity list – Activity attributes – Activity resource Requirements – Resource calendars – Project Scope statement – Risk register – Resource breakdown structure – Enterprise environmental factors – Organizational process assets	– Activity duration estimates – Project document updates	– Experts judgment – Analogous estimating – Parametric estimating – Three-point estimating – Group decision making techniques – Reserve analysis
Develop Schedule	– Schedule management plan – Activity list – Activity attributes – Project schedule network diagrams – Activity resource requirements – Resource calendars – Activity duration estimates – Project scope statement – Risk register – Project staff assignment – Resource breakdown structure – Enterprise environmental factors – Organizational process assets	– Schedule baseline – Project schedule – Schedule data – Project calendar – Project management plan updates – Project document updates – Release and iteration plan updates	

1.3.4 Cost Management

Project cost management includes the processes involved in planning, estimating, budgeting, financing, funding, managing, and controlling costs so that the project can be completed within the approved budget (PMI 2013a). In other words, this knowledge area includes processes to ensure that the project is completed on budget. Cost management for software projects includes making initial estimates and

updating them periodically and may include identifying and forecasting the cost of maintaining and evolving a software product plus listening or updating commercially acquired components over many years (PMI 2013b). All the respective processes with their inputs, outputs, and tools and techniques (to be) used are summarized in Table 1.2.

Table 1.2 Project cost management overview of processes (PMI 2013b)

Process Name	Inputs	Outputs	Tools and Techniques
Plan cost management	– Project management plan – Project charter – Enterprise environmental factors – Organizational process assets	– Cost management plan – Accuracy of estimate – Units of measure – Cost performance measurement methods	– Expert judgment – Analytical techniques – Meetings
Estimate costs	– Cost management plan – Human resource management plan – Scope baseline – Project schedule – Risk register – Enterprise environmental factors – Organizational process assets – Software size and complexity – Rate of work	– Activity cost estimates – Basis of estimates – Project document updates	– Experts judgment – Analogous estimating – Parametric estimating – Bottom-up estimating – Three-point estimates – Reserve analysis – Cost of quality – Project management software – Vendor bid analysis – Group decision making techniques – Time-boxed estimating – Function point and source line of code estimating – Story point and use case point estimating – Estimating reusable code effort – Price to win
Determine budget	– Cost management plan – Scope baseline – Activity cost estimates – Basis of estimates – Project schedule – Resource calendars – Risk register – Agreements – Organizational process assets	– Cost baseline – Project funding requirements – Project document updates	– Cost aggregation – Reserve analysis – Experts judgment – Historical relationships – Funding limit reconciliation

Table 1.2 (continued)

Process Name	Inputs	Outputs	Tools and Techniques
Control costs	– Project management plan – Project funding requirements – Work performance data – Organizational process assets	– Work performance information – Cost forecast – Change requests – Project management plan updates – Project document updates – Organizational process assets updates	– Earned value management – Forecasting – To-complete performance index – Performance reviews – Project management software

1.3.5 Quality Management

Project quality management includes the processes and activities of the performing organization that determine quality policies, objectives, and responsibilities so that the project will satisfy the needs for which it was undertaken. Project quality management uses policies and procedures to implement, within the project's context, the organization's quality management system and supports continuous process improvement activities as undertaken on behalf of the performing organization (PMI 2013a). This area also includes developing plans to ensure that project requirements, including product requirements, are met and validated. Quality management can ultimately establish a quality policy, help understand quality principles introduced by quality experts, develop quality assurance processes, and control the quality of all project deliverables.

Software quality has been a fundamental issue from the early days of developing algorithms. Software quality models include process quality, internal and external product quality, quality in use, data quality, and quality of the software code. The complexities of software quality have led to a number of quality models such as those in ISO/IEC 25000 and the other standards (PMI 2013b).

1.3.6 Human Resource Management

Project Human Resource Management includes the processes that organize, manage, and lead the project team. This knowledge area includes all of the processes used to develop, manage and put the project team together. This involves identifying project stakeholders, developing the project team, motivating the team, understanding management styles, and organizational structure.

Software project staffs collaborate to solve novel problems with incomplete information. Software project managers usually put less emphasis on directing the work and more on facilitating the efficiency and effectiveness of project teams and solving the fitness problem of each team member within the team is critical due to interaction and communication needs of software projects. All the respective pro-

cesses with their inputs, outputs, and tools and techniques (to be) used are summarized in Table 1.3.

Table 1.3 Project human resource management overview of processes (PMI 2013b)

Process Name	Inputs	Outputs	Tools and Techniques
Plan human resource management	– Project management plan – Activity resource requirements – Enterprise environmental factors – Organizational process assets	– Human resource management plan	– Organization charts and position descriptions – Networking – Organizational theory – Expert judgment – Meetings
Acquire project team	– Human resource management plan – Enterprise environmental factors – Organizational process assets	– Project staff assignments – Resource calendars – Project management plan updates	– Pre-assignment – Negotiation – Acquisition – Virtual teams – Multi criteria decision analysis
Develop project team	– Human resource management plan – Project staff assignments – Resource calendars	– Team performance assessments – Enterprise environmental Factors updates	– Interpersonal skills – Training – Team-Building activities – Ground rules – Colocation – Recognition and rewards – Personnel assessment tools – Additional tools and techniques
Manage project team	– Human resource management plan – Project staff assignments – Team performance assessments – Issue log – Work performance reports – Organizational process assets	– Change requests – Project management plan updates – Project documents updates – Enterprise environmental factors updates – Organizational process assets updates	– Observation and Conversation – Project performance appraisals – Conflict management – Interpersonal skills – Additional consideration

1.3.7 Communications Management

Project Communications Management includes the processes that are required to ensure timely and appropriate planning, collection, creation, distribution, storage, retrieval, management, control, monitoring, and the ultimate disposition of project information (PMI 2013a). This knowledge area determines what information is needed, how that information will be sent and managed, and how project perfor-

mance is reported. This involves planning for and distributing information correctly and to the appropriate stakeholders, performance reporting, managing stakeholders, and developing processes to ensure effective transfer of information. Communications management is mainly about effective communication among those involved in the project, from stakeholders to internal project interests at different levels and with different views. The project manager and stakeholders at different levels and with different views and includes developing an understanding of the communications sender–receiver model.

The role of project communication is a primary consideration for software projects, because teams of individuals who engage in closely coordinated, intellectual activities develop software. With no physical product to reference, effective communication is paramount for keeping team members productively engaged and stakeholders informed (PMI 2013b). All the respective processes with their inputs, outputs, and tools and techniques (to be) used are summarized in Table 1.4.

Table 1.4 Project communications management overview of processes (PMI 2013b)

Process Name	Inputs	Outputs	Tools and Techniques
Plan communication management	– Project management plan – Stakeholder register – Enterprise environmental factors – Organizational process assets	– Communication management plan – Project documents update	– Communication requirements analysis – Communication technology – Communication models – Communication methods – Meetings
Manage communication	– Communication management plan – Work performance reports – Enterprise environmental factors – Organizational process assets – Release and iteration plans	– Project communications – Project management plan updates – Project documents updates – Organizational process assets updates – Special communication tools – Update information radiators	– Communication technology – Communication models – Communication methods – Information management systems – Performance reporting – Information radiators – Velocity – Historical velocity – Online collaboration tools

Table 1.4 (continued)

Process Name	Inputs	Outputs	Tools and Techniques
Control communication	– Project management plan – Project communications – Issue log – Work performance data – Organizational process assets – Prioritized backlog – Velocity statistics and projections	– Work performance information – Change requests – Project management plans updates – Organizational process assets updates – Iteration and release plan updates – Reprioritized backlog	– Information management systems – Expert judgment – Meetings – Considerate communications – Automated systems

1.3.8 Risk Management

Project Risk Management includes the processes of conducting risk management planning, identification, analysis, response planning, and controlling risk on a project. The objectives of project risk management are to increase the likelihood and impact of positive events, and decrease the likelihood and impact of negative events in the project (PMI 2013a). This area includes identifying potential project risk events, using qualitative and quantitative analysis to prioritize potential risks, respond to risk situations, and develop risk monitoring and controlling processes. This area can define as a proactive approach to risk management in which the project team and the project manager actively discuss potential risk situations will make the difference between a smooth-flowing project and a project filled with surprises and potential disasters.

Each software development project has different uncertainties and risks, because each project is a unique combination of requirements, design, and construction, resulting in distinct software products (uncertainty arises from a lack of information; risk is a potential issue). Software risk management aims to improve the probability of achieving the project goals; software opportunity management aims to exceed the project goals. Opportunity management is commonly applied in software project management, especially in adaptive projects that have the opportunity to respond to customer-requested changes, apply new technology, or receive additional resources (PMI 2013b). All the respective processes with their inputs, outputs, and tools and techniques (to be) used are summarized in Table 1.5.

Table 1.5 Project risk management overview of processes (PMI 2013b)

Process Name	Inputs	Outputs	Tools and Techniques
Plan risk management	– Project management plan – Project charter – Stakeholder register – Enterprise environmental factors – Organizational process assets	– Risk management plan	– Analytical techniques – Expert judgment – Meetings – Additional consideration
Identify risks	– Cost Management Plan – Schedule management plan – Quality management plan – Human resource management plan – Scope baseline – Activity cost estimates – Activity duration estimates – Stakeholder register – Project documents – Procurement documents – Enterprise environmental factors – Organizational process assets – Risk taxonomies	– Risk register	– Documentation reviews – Information gathering techniques – Checklist analysis – Assumptions analysis – Diagramming techniques – SWOT analysis – Expert judgment – Retrospective meetings
Perform qualitative risk analysis	– Risk management plan – Scope baseline – Risk register – Enterprise environmental factors – Organizational process assets	– Project documents updates	– Risk probability and impact assessment – Probability and impact matrix – Risk data quality assessment – Risk categorization – Risk urgency assessment – Expert judgment – Additional consideration

Table 1.5 (continued)

Process Name	Inputs	Outputs	Tools and Techniques
Perform quantitative risk analysis	– Risk management plan – Cost management plan – Schedule management plan – Risk register – Enterprise environmental factors – Organizational process assets	– Project management plan updates – Project documents update – Additional consideration	– Data gathering and representation techniques – Quantitative risk analysis and modeling techniques – Expert judgment
Plan risk responses	– Risk management plan – Risk register	– Project management plan updates – Project documents update – Additional consideration	– Strategies for negative risks or treats – Strategies for positive risks or opportunities – Contingent response opportunities – Expert judgment – Additional consideration
Monitor & control risks	– Project management pan – Risk register – Work performance data – Work performance reports	– Work performance information – Change Requests – Project management plan updates – Project documents updates – Organizational process assets updates	– Risk reassessment – Risk audits – Variance and trend analysis – Technical performance measurement – Reserve analysis – Meetings

1.3.9 Procurement Management

Project procurement management includes the processes necessary to purchase or acquire products, services, or results needed from outside the project team to complete the project objectives. This includes determining what goods and services should be purchased or developed internally by an organization, planning purchases and, developing procurement documentation such as requests for proposals. It also involves determining appropriate contract types, negotiating terms, selecting sellers, managing contracts through implementation, and then managing project closure and contractual closure.

This knowledge area addresses planning, conducting, controlling, and closing out software project procurements. It also addresses the acquisition of com-

mercially available software for use on a software project. Licensing of software packages, obtaining rights to modify open source software, reuse of existing components, and the purchase of specialty services to build software are all elements of software procurement. Software may also be procured as a service. Just as with commercially available software, it is important to understand the exact nature of the services provided; how they might evolve over time; and what control the customer retains over the data provided to be processed under the service, the results obtained, and any security obligations. These considerations are usually covered in a service level agreement. Often, the standard agreement issued by the provider may not meet the acquirer's specific needs (PMI 2013b).

1.3.10 Stakeholder Management

Project stakeholders include anyone influenced by or influencing the result of a (software) project. The involvement of different types of customers, developers, management, shareholders, competitors as well as standards and legislation are important for the execution and success of the project. Project stakeholder management includes the processes to ensure identification of stakeholders and to plan, manage and control their engagement which is required to identify the people, groups, or organizations that could impact or be impacted by the project, to analyze stakeholder expectations and their impact on the project, and to develop appropriate management strategies for effectively engaging stakeholders in project decisions and execution. Stakeholder management also focuses on continuous communication with stakeholders to understand their needs and expectations, addressing issues as they occur, managing conflicting interests and fostering appropriate stakeholder engagement in project decisions and activities.

Stakeholder management is critical for achieving successful outcomes for software projects because software has no physical presence and is often novel. Software is difficult to visualize until it is demonstrated. In addition, there often exists a gulf of expectation between what a customer or product owner states and what the developer interprets. Misalignments among stakeholders represent a major risk for successful completion of software projects (PMI 2013b).

1.4 The Book's Coverage of the PMBOK Knowledge Areas

In Table 1.6, we define each chapter's main scope related to the established PMBOK knowledge areas. It shows that the collection of chapters is in good match with the project management knowledge areas introduced by the PMBOK (PMI 2013a). Some of the chapters are relate to more than one knowledge area. Chapter 8 of this book is the only exception on that. This chapter is more related to "The Standard for Portfolio Management" (PMI 2013c), published recently in response to the increasing acceptance of portfolio management. On the other hand, almost all knowledge areas are covered by at least one chapter, with most of them having

multiple connections. Again, there is one exception, and this is the area of procurement management, being outside the scope of this book.

Table 1.6 Book chapters vs. PMBOK knowledge areas

Chapter	Title	PMBOK knowledge areas
2	Rethinking Success in Software Projects: Looking beyond Failure Factors	– Communications Management – Time Management – Cost Management – Scope Management – Risk Management – Stakeholder Management
3	Cost Prediction and Software Project Management	– Cost Management
4	Human Resource Allocation and Scheduling for Software Project Management	– Human Resource Management – Time management
5	Software Project Risk and Opportunity Management	– Risk Management – Cost Management
6	Model-based Quality Management of Software Development Projects	– Quality Management
7	Supporting Project Management through Integrated Management of System and Project Knowledge	– Integration Management – Scope Management – Time Management – Quality Management – Communications Management – Risk Management – Stakeholder Management
8	A Framework for Implementing Product Portfolio Management in Software Businesses	– Not in the domain of PMBOK knowledge areas, but related to Portfolio Management
9	Managing Global Software Projects	– Communications Management – Human Resource Management
10	Motivating Software Engineers Working in Virtual Teams Across the Globe	– Human Resource Management
11	Agile Project Management	– Time Management – Human Resource Management – Communications Management – Integration Management – Stakeholder Management
12	Distributed Project Management - Ten Misconceptions That Might Kill Your Distributed Project	– Communications Management – Time Management – Cost Management – Human Resource Management

Table 1.6 (continued)

Chapter	Title	PMBOK knowledge areas
13	Management and Coordination of Free/Open Source Projects	– Human Resource Management – Communications Management – Integration Management
14	Inner Source Project Management	– Communications Management – Integration Management – Time Management – Cost Management
15	Search-Based Software Project Management	– Human Resource Management – Time Management – Cost Management
16	Social Media Collaboration in Software Projects	– Communications Management – Stakeholder Management
17	Process Simulation – A Tool for Software Project Managers?	– Time Management – Cost Management – Risk Management
18	Occam's Razor and Simple Software Project Management	– Cost Management

1.5 The Multi-disciplinary Nature of Project Management

Project Management (PM) is seen by Kwak and Anbari (2008) as the integration and application of what was called *allied disciplines*. The authors analyzed past, current and future trends of the allied disciplines by exploring, identifying and classifying top management journal articles related to project management research. The authors defined eight main areas (allied disciplines) and conducted some research trend analysis. They studied 537 articles published between 1950 and June 2007. A ranking of the disciplines according to the number of studies is presented in Table 1.7.

Kwak and Anbari (2008) predicted that project management becomes more multidisciplinary and flexible in adopting tools from other disciplines. As documented by the book, this also applies to software project management.

Accurate planning and estimation of cost and schedule is difficult for all kinds of projects but it is particularly difficult for software projects because cognitive nature of work, wide variety of productivity among individuals, poor requirement definition, data inaccuracy of past projects. In the study of allied disciplines, the authors predict that *Strategy/Integration/Portfolio Management/Value of Project Management and Marketing* and *Quality Management/Six Sigma/Process Improvement* should have a growing impact on project management, as business strategies are

Table 1.7 Ranking of PM allied disciplines according to Kwak and Anbari (2008)

Rank	Discipline	Percentage of publications
1	Strategy/Integration/Portfolio Management/Value of Project Management and Marketing	30
2	Operations Research/Decision Sciences/Operation Management/ Supply Chain Management	23
3	Organizational Behavior/Human Resources Management	13
4	Information Technology/Information Systems	11
5	Technology Applications/Innovation/New Product Development/ Research and Development	11
6	Performance Management/Earned Value Management/Project Finance and Accounting	7
7	Engineering and Construction/Contracts/Legal Aspects/Expert Witness	3
8	Quality Management/Six Sigma/Process Improvement	2

developed and qualities are measured and analyzed to plan and implement effective project management. Chapters 6, 7 and 8 are contributions in this direction.

According to (PMI 2013b), creation of software requires innovative problem solving to create unique solutions. Software projects are more akin to research and development projects than to construct and manufacturing projects. From an allied disciplines point of view, *Technology Applications/Innovation/New Product Development/Research and Development* as well as *Performance Management/Earned Value Management/Project Finance and Accounting* are poised to make major breakthroughs given the recent organizational interest and institutional determination on achieving project success (Kwak and Anbari 2008). Chapters 2, 3 and 5 are contributions in this direction.

Operations Research/Decision Sciences/Operation Management/Supply Chain Management, Performance Management/Earned Value Management/Project Finance and Accounting, Information Technology/Information Systems, and Technology Applications/Innovation/New Product Development/Research and Development will work together to deliver tools and techniques to allow the ‘‘science’’ of planning, scheduling, and cost control to function in a real project delivery environment. Chapters 4, 15, 17 and 18 are contributions in this direction.

1.6 The Future of Software Engineering

Boehm predicted that in response to the increasing criticality of software within systems and the increasing demands being put onto 21st century systems, systems and software engineering processes will evolve significantly over the next two decades (Boehm 2006a). By analyzing today’s trends, Goff stated that the average en-

terprise in 2025 even with innovation, market pressures and significant effort, will still be struggling to adopt with today's leading technologies (Goff 2009).

In the case of predicting the future in software project management both trends of project management and software engineering could be helpful. The ten trends in software engineering discussed in (Boehm and Lane 2010) are:

1. Increasing emphasis on rapid development and adaptability;
2. Increasing software criticality and need for quality assurance;
3. Increased complexity, global systems of systems, and need for scalability and interoperability;
4. Increased needs to accommodate Commercial-Off-The-Shelf (COTS), software services, and legacy systems;
5. Increasingly large volumes of data and ways to learn from them;
6. Increased emphasis on users and end value;
7. Computational plenty and multicore chips;
8. Increasing integration of software and systems engineering;
9. Increasing software autonomy; and
10. Combinations of biology and computing

Technology trends and continuing need for product differentiation, globalization and its effect on market and processes for new technology introduction accelerate system change, which makes a trade of between development speed and costs of development a necessity. Nidiffer and Dolan (2005) stated that ever increasing growth and complexity of software-intensive systems and appearance of geographically distributed systems are today's trends. Geographically distributed projects let managers compress schedules by employing larger workforces than could fit in a single location, using time zone differences to increase the number of productive work hours in a day, and securing scarce resources such as knowledge experts and other specialized resources no matter where they reside. However, these benefits come with increased risks because of the lack of face-to-face communication, in particular, the potential loss of trust, collaboration, and communication richness.

On the other hand, as a probable situation computational plenty will spawn new types of applications and platforms. These will present process-related challenges for specifying their configurations and behavior; generating the resulting applications; verifying and validating their capabilities such as performance, and dependability; and integrating them into even more complex systems of systems. All these may need minor changes in the domain of project management. (Boehm 2006b) indicates that key challenges include cross-cultural bridging; establishment of common shared vision and trust; contracting mechanisms and incentives; handovers and change synchronization in multi-time zone development; and culture-sensitive collaboration-oriented groupware.

1.7 Software Project Management - Past and Future

While the era of modern project management as a discipline started in the 1950's, it was by about 1980 that software development industry started implementing some established project management practices. During this period, Barry Boehm defined the whole field of software engineering economics with introducing COCOMO, the Constructive Cost Model for software (Boehm 1981). In 1984, The *Project Management Institute* PMI was offering its certification program for the first time. By the 1990's, project management theories and practice accepted widely and project management was known as a profession.

More recently, project management has become a factor present at all levels, in all divisions of many companies. (Goff 2009) discussed visions for the project management software industry from an industrial point of view. The following trends were projected:

- Project portfolio management
  Currently there are several vendors which create tools to support portfolios in the case of managing integration and it should support future of project management.
- Collaboration with the means of virtual teams and social networks
  Combination of virtual teams with social networks in conjunction with more advanced tools supporting collaboration and communication creates new opportunities of working together as virtual teams, not being co-located. This is projected to result in a massive increase in the market size for PM software.
- Mastery of real time tracking
  Time tracking based on timesheet is a critical success factor for many teams. Earned value management becomes more useful as the window of currency moves from one month after the work only one day and much accurate and reliable information is needed.
- Capture and reuse of project knowledge
  Developing project schedule and project plan once would be really great in fact all successive projects reuse the materials as templates. This reuse could be applied in project documents such as test plans or requirements, too.
- Dashboards and project intelligence
  Project tracking may base on easy to measure trailing indicators instead of critical indicators. Monitoring project status in an efficient way will help in project communication improvement and smarter decisions.
- Project management absorbed into enterprise systems
  More and more overlap with other enterprise system functionality, all relying on enterprise data. Strongest ties are to Enterprise Resource Planning ERP systems, but links and increasing degree of overlap also exists to Customer Relationship Management CRM, Supply Chain Management SCM, and Product Life-cycle Management PLM systems.

1.8 This Book

Software Project Management in a Changing World is not "just another book" in the area of software project management. It brings together the various current directions within the discipline, which are so far presented mainly in individual articles or books. Whenever appropriate, the content of the book is based on evidence coming from empirical evaluation of the proposed approaches.

There is already a great variety of books and publications offering guidelines and best practices. To keep the content of the book focused, the implicit assumption here is that we do not address small-scale projects. For professionals, the book is intended to be a source of inspiration to refine their project management skills into new areas. For researchers and graduate students, the book presents some of the most recent methods and techniques to accommodate the new challenges of the discipline. The goal of the book is to find a good balance between new results and putting together existing material to allow its usage in new contexts.

The book consists of four parts, preceded by a general introduction. Each of the parts consists of a sequence of chapters. All the four parts start with a brief overview outlining its content.

Introduction

1. Software Project Management: Setting the Context by Günther Ruhe and Claes Wohlin

Part I – Fundamentals

2. Rethinking Success in Software Projects: Looking Beyond the Failure Factors by Darren Dalcher
3. Cost Prediction and Software Project Management by Martin Shepperd
4. Human Resource Allocation and Scheduling for Software Project Management by Constantinos Stylianou and Andreas S. Andreou
5. Software Project Risk and Opportunity Management by Barry Boehm

Part II – Supporting Areas

6. Model-based Quality Management of Software Development Projects by Jens Heidrich, Dieter Rombach and Michael Kläs
7. Supporting Project Management through Integrated Management of System and Project Knowledge by Barbara Paech, Alexander Delater and Tom-Michael Hesse
8. A Framework for Implementing Product Portfolio Management in Software Businesses by Erik Jagroep, Sjaak Brinkkemper, Inge van de Weerd and Ton Dobbe
9. Managing Global Software Projects by Christof Ebert

Software Project Management is a dynamically evolving discipline which constitutes of a wide range of sub-disciplines. Each of them is covering specific practices, methods and tools. Even though we covered a broad range of topics, the book is not intended to be comprehensive. The selection for the chapters made was primarily based on perceived importance of the topics, although being impossible to cover all topics of interest. For example, even though there exists a large variety of proprietary and open source tools (Pereira et al. 2013), we considered software project management tools in general being outside the scope of the book. Also, some recent trends such as the one of utilizing the power of predictive analytics (Hassan 2013) or of visualization of project data (Novais et al. 2013) for the purpose of qualifying project management were not included.

The book is intended to attract readers from both academia and practice. The targeted benefits for the readers are:

- Getting an overview of most recent methods and techniques,
- Support for better decision-making and provide inspiration when conducting the activities related to project management in new contexts, and
- Learn about most recent trends and understanding the implications of their implementations.

References

Basili VR, Shull F, Lanubile F (1999) Building knowledge through families of experiments. IEEE Transactions on Software Engineering Vol. 25 (4):456-473

Boehm BW (1981) Software Engineering Economics. Pentice Hall

Boehm BW (2006a) Some future trends and implications for systems and software engineering processes. Systems Engineering Vol. 9 (1):1-19

Boehm BW (2006b) A view of 20th and 21st century software engineering. 28th International Conference on Software Engineering, pp 12-29

Boehm BW, Lane JA (2010) Evidence-based software processes. International Conference on Software Process, ICSP 2010, LNCS vol 6195, pp 62-73

Brooks FP (1987) No Silver Bullet: Essence and Accidents of Software Engineering. IEEE Computer Vol. 20 (4):10-19

Goff SA (2009) Visions For the Project Management Software Industry. In: Cleland D, Bidanda B (eds) Project Management Circa 2025. Project Management Institute, Athabasca University pp 1-15

Hassan AE (2013) Software Analytics: Going beyond Developers. IEEE Software 30 (4):53

Kruchten P (2011) The Frog and the Octopus – A Model of Software Development. CSI Communications Vol. 35 (4):12-15

Kwak YH, Anbari FT (2008) Impact on Project Management of Allied Disciplines: Trends and Future of Project Management Practices and Research, Project Management Institute

Larman C, Basili VR (2003) Iterative and incremental development: A brief history. Computer Vol. 36 (6):47-56

Nidiffer KE, Dolan D (2005) Evolving distributed project management. IEEE Software 22 (5):63-72

Novais RL, Torres A, Mendes TS, Mendonça M, and Zazworka N (2013) Software evolution visualization: A systematic mapping study. Information and Software Technology 55, 1860-83

Pereira AM, Goncalves RQ, von Wangenheim CG, and Buglione L (2013) Comparison for open source tools for project management. International Journal of Software Engineering and Knowledge Engineering 23, 189-209

PMI (2013a) A guide to the project management body of knowledge (PMBOK® guide). Fifth edn. Project Management Institute

PMI (2013b) Software Extension to the PMBOK Guide. Fifth edn. IEEE Computer Scociety, Project Management Institute

PMI (2013c) The Standard for Portfolio Management. Third edn. Project Management Institute

Author Biography

Günther Ruhe holds an Industrial Research Chair in Software Engineering at University of Calgary. Dr. Ruhe received a doctorate rer. nat degree in Mathematics with emphasis on Operations Research from Freiberg University and a doctorate habil. nat. degree (Computer Science) from University of Kaiserslautern. From 1996 until 2001, he was the deputy director of the Fraunhofer Institute for Experimental Software Engineering Fh IESE. Ruhe received an iCORE research award for the period from 2001 to 2007. Since 2007, he serves as an Associate Editor of the journal of Information and Software Technology, published by Elsevier. His main research interests are in the areas of Product Release Planning, Software Project Management, Empirical Software Engineering as well as Search-based Software Engineering. He is a Senior member of IEEE and a member of the ACM. Dr. Ruhe is the Founder and CEO of Expert Decisions Inc., a University of Calgary spin-off company created in 2003.

Claes Wohlin is Professor of Software Engineering at Blekinge Institute of Technology, Sweden. From January 1, 2014 he is the Dean for the Faculty of Computing Professor Wohlin is a guest professor at Shandong University at Weihai in China. Prior to joining BTH in 2000 he held professor chairs at Lund and Linköping Universities. He has been a Visiting Professor at Chalmers University of Technology in Göteborg (2005-2008) and at the University of New South Wales in Sydney, Australia (2009-2011). Professor Claes Wohlin received a Ph.D. degree in Communication Systems from Lund University in 1991. Since January 2008, Professor Wohlin is Editor-in-Chief of the journal of Information and Software Technology, published by Elsevier, and he has been the Co-Editor-in-Chief of the journal since 2001. In 2011, Claes Wohlin was elected member of the Royal Swedish Academy of Engineering Sciences. He is a Senior member of IEEE.

Part I — Fundamentals

Introduction

Software project management as such implies managing a set of fundamental aspects in relation to the development and evolution of software as outlined in Chapter 1. In this part of the book, we have invited some of the leading experts in relation to a set of fundamental areas for a software project manager to master, and they share their knowledge, insights and accompanying recommendations and conclusions in four chapters in this part of the book.

In Chapter 2, Darren Dalcher challenges us to rethink the definition of software project management success. The chapter starts off by summarizing some of literature in reporting on the failures in relation to software projects. The data clearly indicates that it is a daunting task to succeed with your software project in particular if it is a large project. He goes on to explain how the project manager all too often is not really in charge of the main success factors: software performance in terms of what the software should achieve, cost for development and the delivery time. Dalcher describes the need to have multiple categories for success, for example project success and product success. Based on this reasoning, the chapter presents a four level model of success in relation to software projects and their output. Some examples are presented to highlight how the perception of failure or success may change with the levels as well as over time.

Martin Shepperd provides a review of software project cost prediction in Chapter 3. He starts by discussing some of the main reasons for the problems in relation to cost prediction: complexity of the development, software development is a design activity, estimates are needed early and the development is often put under social and political pressure. The chapter continues by reviewing some of the techniques for cost prediction both formal models and expert judgment. Challenges in relation to both techniques for cost prediction are discussed along with the need to consider both the people and formal aspects. Particular emphasis is given to the problems that arise from cognitive biases, that is heuristics that cause even experts to deviate from optimal or logical decision-making. Based on the discussion, Shepperd provides some recommendations in relation to improving the practice of software cost prediction.

In Chapter 4, Constantinos Stylianou and Andreas S. Andreou address some of the issues in relation to human resource allocation and scheduling in software projects. Given that software development is a design activity, the human aspect of development and management is crucial. Stylianou and Andreou focus on human resources from a planning perspective, including assigning developers and teams to tasks within the development. The chapter provides an overview of some recent approaches to human resource allocation and scheduling. The most common gen-

eral approaches in research relate to using different specific techniques in relation to mathematical modelling and computational intelligence. The chapter provides summaries and references to specific approaches to human resource allocation and scheduling. The chapter ends with a discussion on the shift towards also incorporating non-technical factors in human resource allocation and scheduling and, in particular, the adoption of a more human-centric approach by using software developers' personality types to allocate tasks and form teams.

Barry Boehm has authored the final chapter (Chapter 5) in this part. The chapter discusses risk and opportunity management in relation to software projects. Risk is an uncertain condition and event, which may jeopardize the success of the software project. Boehm describes a number of aspects to be addressed to manage risk. He discusses the duality between risk and opportunity, where risks may generate losses and opportunities may result in gains. He describes and illustrates how risk and opportunity exposure can be managed. Boehm highlights that not only risks must be managed, but also the opportunities. He continues with discussing the joint management of risks and opportunities. Furthermore, he introduces the concept of lean risk management plans and discusses risk tracking.

The four chapters in this part give an in-depth insight into some of the challenges related to a selection of fundamental areas for anyone conducting research into software project management or managing a software project.