



SOFTWARE PROCESS

Assignment 1

April 20, 2018

Student:
Piotr Kosytorz
UvAnetID 11876964

Tutor:
drs. Hans Dekker

Course:
Software Process

Contents

1	Best practices	1
2	Methods	2
2.1	Waterfall (1970)	2
2.2	Scrum (1986)	3
2.3	Agile (2001)	4
2.4	RUP (2003)	5
2.5	Extreme Programming (2000)	5
2.6	DevOps (2008)	6
3	Discussion	6

1 Best practices

In an article from 2000 by James Salapatras for the Project Management Institute, nine best practices for achieving success in a project are provided[3]:

1. Defined Life Cycle and Milestones
Organizations need to map and define phases, deliverables, key milestones and sufficiency criteria for each group involved in the project.[3]
2. Stable Requirements and Scope
Effective project management requires that project requirements, objectives and scope be documented and become stabilized at some point early in the project life cycle.[3]
3. Defined Organization, Systems, and Roles
In any organization projects must have defined roles for the project manager, functional managers and project team members. Accountabilities must be identified for all.[3]
4. Quality Assurance
Quality on projects requires the identification of standards and criteria to be set in each phase of the project life cycle for both the product and the process. Quality means making and meeting agreed to commitments with a constant eye for improvement.[3]

5. Planned Commitments

Plans must be based upon the process capability of the organization and not upon wishful thinking. It is common to see wishful project schedules built upon a “house of cards” where sufficient resources are not available. Plans must be more than schedules in that they address all nine elements of the project management process.[3]

6. Tracking and Variance Analysis

Projects should be managed using an exception process in which deviations from plans are reported and resolved.[3]

7. Corrective Action Decisions

When variances from plan are detected, the default assumption is that the team or functional groups will work to put the project back on track. Without a clear procedure corrective action can have many outcomes, not all consistent with corporate objectives.[3]

8. Escalation and Issue Management

An effective escalation procedure requires issues and problems to be worked first by the lowest appropriate level. If the issue cannot be resolved and closed, then it must be elevated to the next highest organizational level, and so on until the issue is closed. [3]

9. Work Authorization and Change Control

Late changes in projects are a major source of disruption that lead to schedule slippage, cost overruns, insertion of defects and rework. A formal system of change control and change management must be in place. Changes caused by scope creep must be resisted and change control is needed to prevent these problems.[3]

In my report on Methods, I will be referring to these nine best practices.

2 Methods

2.1 Waterfall (1970)

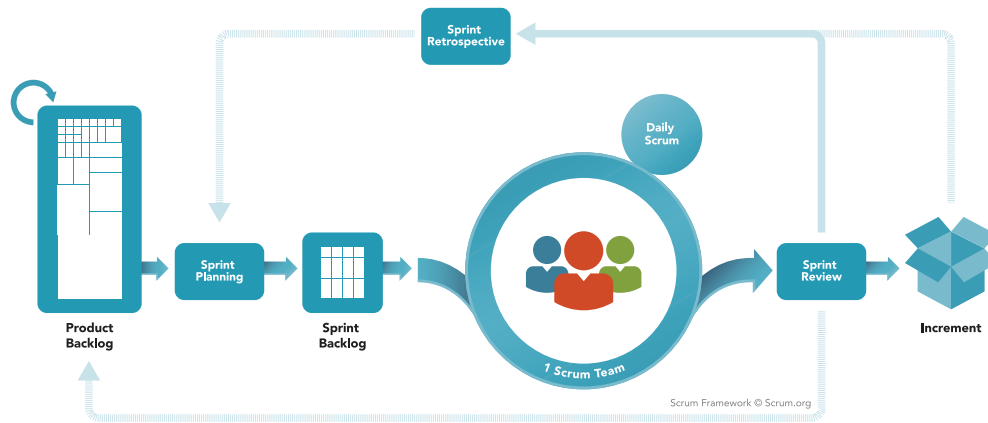
Waterfall is a sequentially structured project management methodology. According to Royce [7], the methodology can be described as a sequence of the following phases:

1. System requirements
2. Software requirements
3. Analysis
4. Program design
5. Coding
6. Testing
7. Operations

Royce draws attention to two very important aspects of waterfall methodology in its starting phases: documentation and project knowledge. As he says, *During the early phase of software development the documentation is the specification and is the design.[7]* Documentation is thus the backbone of project management. Writing documentation guarantees early testing of (errors) in the design and concept: *Without good documentation every mistake, large or small, is analysed by one man who probably made the mistake in the first place because he is the only man who understands the program area[7]*

Figure 1: The Scrum Framework[9]

SCRUM FRAMEWORK



The second thing Royce mentions is project knowledge: there should be at least one person that has a deep understanding of the whole system. This knowledge must be the exhaustively put into documentation. Each project phase must deliver appropriate documentation: *At least one person must have a deep understanding of the system which comes partially from having had to write an overview document*[7]

Waterfall as methodology sets up a clear structure in project management. Each proceeding phase is a logical result of the previous one.

Waterfall seems a good solution to projects which requirements can be fully determined from the beginning. Unfortunately, there is very little (or actually none) space for requirements change during the project. Royce gives examples of systems that take 12 months to develop.

Due to the fact that in waterfall, proceeding from one phase another requires an extended amount of work and documentation, every change in requirements would heavily disturb the process.

Waterfall is a good fit for highly specialized projects, that stay conservative in their requirements and don't have much rotation of personnel in their development teams.

Another aspect in favour of the statement that Waterfall is more a fit for projects that are conservative in their requirements is the fact, that Royce mentions: *If the computer program in question is being developed for the first time, arrange matters so that the version finally delivered to the customer for operational deployment is actually the second version insofar as critical design/operations areas are concerned.*[7]

According to the list of best practices from chapter 1, waterfall is missing Corrective Action Decisions and Work Authorization and Change Control, as waterfall is designed to be sequential, and there is no formal mechanism for re-iterating parts of the project.

2.2 Scrum (1986)

According to Schwaber[8] *Scrum is a framework for developing and sustaining complex products.*[8]

Scrum consists the following elements:

- Product: *what has to be produced*
 - Backlog: *list of elements that will eventually collectively produce the end product*
- Sprint: *a sub-project limited by time with a precisely determined output*
 - Planning
 - Backlog: *list of tasks in current sprint (a subset of product backlog)*
 - Review: *analysis of the meaning and fitness of the current sprint in relation to the whole product*
 - Retrospective: *analysis of what have we done wrong, what can we do better next time*
- Scrum team
 - Product Owner: *Domain specialist*
 - Development Team *The blue collar workers*
 - Scrum Master: *Judge in the ring: makes sure that there is a healthy relationship between the product owner and the development team, and that they follow the rules.*
- Daily scrum: *a 15-minute time-boxed daily meeting/discussion*
- Increment: *A finished spring that builds up the final product.*

Scrum introduces a lot of flexibility in project management and development. The work is divided into manageable chunks. There is always someone who knows what has to be achieved in the project (Product Owner), and someone who takes care of the team and makes sure that the rules are being played fair (Scrum Master).

Scrum can be applied to both: big and small projects. It allows project adaptation, thus change in requirements is taken into account.

In the terms of best practices introduced in chapter 1, scrum fulfils all the criteria, if done correctly. In practice Stable Requirements and Scope can be a weak point, because as every Agile method, scrum puts pressure more on single iteration (sprints) than on a holistic vision of the project.

2.3 Agile (2001)

Agile manifesto[5]:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

Agile as a philosophy puts pressure on the value of human and practical aspects of software development, rather than of a formalized path to follow.

Salapatas' best practices are not applicable here.

2.4 RUP (2003)

Rational Unified Process is another software development framework with another look into the problem of project development. The main focus of the framework is to ensure that everyone in the team knows what to do, therefore it is based on three elements:

- Roles (who)
- Work products (what)
- Tasks (how)

Shortly: **Who** produces **what**, and **how** does he do that?[13][6]

On the other hand, RUP tries to apply the best of both: Waterfall and Scrum. It consists of four phases:

- Inception,
- Elaboration,
- Construction,
- Transition,

where all six “engineering disciplines” are present, but with different intensity according to different phases.

Within all the phases RUP introduces international development.

RUP seems like a good solution for big and highly skilled teams of professionals within big organizations (like IBM itself).

RUP implements all of Salapatas’ best practices.

2.5 Extreme Programming (2000)

Extreme Programming is an Agile process, that focuses on:[2][12][1]

- delivering software (features) as soon as possible,
- testing/gaining feedback from the customers as soon as possible,
- delivering what you need when you need it, and only when you need it,
- staying flexible to respond to changing requirements and technology,
- enabling the teams to self-organize around the problem to solve it as efficiently as possible.

Extreme programming seems can be successfully applied in small teams that do not require highly skilled programmers [1]. Start-up projects could benefit from this approach as well, since the request for requirements change there is high. XP, however, does not seem as a solution for big and highly specialized projects that require a holistic and deep understanding of the domain.

In terms of Salapatas’ best practices XP is missing Stable Requirements and Scope and Defined Organization, Systems, and Roles.

2.6 DevOps (2008)

According to Wikipedia: *DevOps (a clipped compound of “development” and “operations”) is a software engineering culture and practice that aims at unifying software development (Dev) and software operation (Ops). The main characteristic of the DevOps movement is to strongly advocate automation and monitoring at all steps of software construction, from integration, testing, releasing to deployment and infrastructure management. DevOps aims at shorter development cycles, increased deployment frequency, and more dependable releases, in close alignment with business objectives.*[11]

DevOps changes the way teams of specialists cooperate with each other. This results in process change.

According to Amazon[10], the following are the DevOps best practices:

- Continuous Integration
- Continuous Delivery
- Microservices
- Infrastructure as Code
- Monitoring and Logging
- Communication and Collaboration

What does DevOps change to Waterfall, Scrum, Agile, RUP, and XP? DevOps introduces automation of repeatable processes, strong decoupling of projects and greater insight in software performance and software process. Linus Torvalds in his talk about Git at Google link said, that the speed of your tools (processes) changes, then the way you use them changes as well. To me this is a great paraphrase of what DevOps does to software development methodologies - it changes the way we apply them. The focus is not so much laid on the interfaces between different phases of different methodologies, but rather on the core problems of software development and software quality itself.

3 Discussion

As Bethany Cartwright aptly states in her article[4]: *There are two main types of project management methodologies: waterfall and agile.* They differ a lot, and when looking at the time and circumstances of their publications, it is safe to say, that waterfall was and still is the way to go for bog projects that can be hold a conservative approach to their requirements. Since the beginning of the IT revolution, bigger and bigger presence of software in business and all kinds of devices, and especially since the Internet, waterfall had to make more and more place for agile. The rate of delivering software has rapidly increased in the last years. In the guest lectures by Amazon and Adyen, we have learned that they deploy their software even multiple times a day. Waterfall cannot be applied in such circumstances any more.

When it comes to all the different Agile methods, like Scrum, XP, etc., the choice for specific method (or framework) is a matter of fitness for a specific project and team.

The only exception is RUP, as a combination of best practices of both: Waterfall and Agile is probably a great alternative for big-scale Waterfall projects that need a transition to a more agile approach.

Regardless of the choice of a method, the process does not guarantee a successful delivery of the end product. All of the discussed methods have their strong and weak sides, and they are only tools to improve the work flow.

Table 1: Methods trade-off

Method/characteristic	Waterfall	Scrum	Agile	RUP	XP	DevOps
Publication year	1970	1986	2001	2003	2000	2008
Project time open/closed	closed	open	n.a.	closed	open	open
Self-organization	no	yes	yes	no	yes	yes
Transparency	no	yes	n.a.	yes	yes	yes
Inspection	no	yes	n.a.	yes	yes	yes
Adaptation	no	yes	yes	yes	yes	yes
Iteration	no	yes	yes	yes	n.a.	yes

References

- [1] Agile Alliance. *What is Extreme Programming (XP)?* Feb. 2018. URL: [https://www.agilealliance.org/glossary/xp/#q=~\(filters~\(postType~\(' post~ ' aa_book~ ' aa_event_session~ ' aa_experience_report~ ' aa_glossary~ ' aa_research_paper~ ' aa_video\)~tags~\(' xp\)\)~searchTerm~ ' ~sort~false~sortDirection~ ' asc~page~1\).](https://www.agilealliance.org/glossary/xp/#q=~(filters~(postType~(' post~ ' aa_book~ ' aa_event_session~ ' aa_experience_report~ ' aa_glossary~ ' aa_research_paper~ ' aa_video)~tags~(' xp))~searchTerm~ ' ~sort~false~sortDirection~ ' asc~page~1).)
- [2] Kent Beck. "Extreme programming: A gentle introduction". In: URL <http://www.extremeprogramming.org> (2004).
- [3] *Best practices—the nine elements to success*. Paper presented at Project Management Institute Annual Seminars and Symposium. Houston, TX. Newtown Square: Project Management Institute, 2000.
- [4] Bethany Cartwright. *The Only 2 Types of Project Management Methodology That Matter*. May 2015. URL: <http://careers.alot.com/planning/the-only-2-types-of-project-management-methodology-that-matter--3610>.
- [5] Martin Fowler and Jim Highsmith. "The agile manifesto". In: *Software Development* 9.8 (2001), pp. 28–35.
- [6] Andrew Powell-Morse. *Rational Unified Process: What Is It And How Do You Use It?* Nov. 2017. URL: <https://airbrake.io/blog/sdlc/rational-unified-process>.
- [7] W W Royce. "Managing the development of large software systems". In: *Electronics* 26.August (1970), pp. 1–9. ISSN: 03784754.
- [8] Ken Schwaber and Jeff Sutherland. "The scrum guide". In: *Scrum Alliance* 21 (2011).
- [9] Alliance Scrum. "What is Scrum?" In: *We've been covering Scrum in such detail it makes you sick. And now* (2013).
- [10] Amazon Web Services. *What is DevOps? - Amazon Web Services (AWS)*. URL: <https://aws.amazon.com/devops/what-is-devops/>.
- [11] Wikipedia. *DevOps — Wikipedia, The Free Encyclopedia*. <http://en.wikipedia.org/w/index.php?title=DevOps&oldid=837064470>. [Online; accessed 20-April-2018]. 2018.
- [12] Wikipedia. *Extreme programming — Wikipedia, The Free Encyclopedia*. <http://en.wikipedia.org/w/index.php?title=Extreme%20programming&oldid=830770686>. [Online; accessed 20-April-2018]. 2018.
- [13] Wikipedia. *Rational Unified Process — Wikipedia, The Free Encyclopedia*. <http://en.wikipedia.org/w/index.php?title=Rational%20Unified%20Process&oldid=835850208>. [Online; accessed 20-April-2018]. 2018.