Agile Softwareentwicklung in der Wissenschaft

Beleuchtung von Fallstudien

Fabian Schacht

Arbeitsbereich Wissenschaftliches Rechnen
Fachbereich Informatik
Fakultät für Mathematik, Informatik und Naturwissenschaften
Universität Hamburg

16.05.2015

Introduction

Case Studies / Main Publications

- Is Scrum and XP suitable for Computational Science and Engineering (CSE) Development? (2012) [Blo12]
- Limitations of Agile Processes 2002 [TFR02] , 2009 [Col09] , 2014/2015 [MVS15]
- What Do We Know about Scientific Software Development's Agile Practices? [SHPL12]

Agenda

1. Introduction
   - The CSE Domain
2. Main Part
   - Agile and the CSE Domain
     - Limitations and Disadvantages of Agile Processes
     - When to use the Agile Model
   - Scrum and the CSE Domain
     - A Detailed Look into Scrum
   - XP and the CSE Domain
   - Practical Examples
3. Summary
4. Bibliography

Introduction

Business Oriented Software Development Methods vs. CSE

Table: Business Oriented Software Development Methods vs. CSE [Blo12]

Commercial SD Methods	CSE
large-scale business oriented projects	large & small-scale scientific projects
long expected life time	some projects with long, most with short expected life time
need maintenance and further development during their life time	written for a particular purpose
developed by a group of people or even many groups	developed by a single researcher or a small group of researchers
large user base	smaller potential user base
development methods used by experienced developers	CSE systems developers on the other hand have often less experience
experienced in development methods	risk getting entangled in details and overhead

Hypothesis

Hypothesis

As the needs from CSE differ from what traditional SD offers, a new approach might be needed or might be beneficial

Chronological Anchor

Chronological Anchor

Agile (Agile Manifesto) introduced in 2001 evolved from lightweight software development methods in the mid-1990s [Wik]

Figure: Taken from www.agilealliance.org

CSE Development & Agile Development

Does the "Agile Manifesto" Support CSE Development?

Customer Satisfaction	●	
Embrace Change	● ⋆	○
Frequent Deliveries	●	
Work Together	●	
Motivated Individuals	● ⋆	×
Face-to-face Conversation	●	×
Working Software	● ⋆	○
Sustainable Development	●	
Technical Excellence	●	×
Simplicity	●	×
Self-organizing Teams	●	×
Self-reflection	●	

less relevant: (no dot) | relevant: ● | extra relevant: ●⋆

Table: Does the "Agile Manifesto" Support CSE Development? [Blo12]

Agile

Interim Conclusion

All points are judged as relevant or extra relevant. Working according to an agile method should hence be beneficial for CSE development. [Blo12]

Limitations of Agile Processes

Stand 2002 [TFR02]
Limited support for:

- subcontracting
- development involving large teams
- distributed development environments
- building reusable artifacts
- developing safety-critical software
- developing large, complex software

Limitations of Agil Processes

Stand 2009 [Col09]

- Agile designed by stars for a team of stars
- Doesn't fit with organizational culture
- Team ownership vs. individual accountability (Reward model)
- Limited support for development involving large teams
- Limited support for distributed development environments
- Almost no focus on methodology (analysis, architecture, implementation, project management, configuration management...)

Limitations of Agil Processes

Stand 2015

- Customer unavailability [MVS15]
- Support for development involving large teams
  - Large-Scale Scrum (LeSS) for large teams (controversial) [Wik]
- Support for distributed development environments
  - Distributed development [Ltd13] (Can you still call it agile?)
- Insufficient requirements gathering [MVS15]
- Turnover Team [MVS15]
- Software Maintenance [MVS15]

Limitations of Agil Processes

Interim Conclusion

Some negative points haven't been addressed over the years

Disadvantages of Agile model

[Cer15]

- Guessing the Required Effort
  In case of some software deliverables, especially the large ones, it is difficult to assess the effort required at the beginning of the software development life cycle.
- Design and Documentation
  There is lack of emphasis on necessary designing and documentation.
- Lack of Focus
  The project can easily get taken off track if the customer representative is not clear what final outcome that they want.
- Lack of Expertise
  Only senior programmers are capable of taking the kind of decisions required during the development process. Hence it has no place for newbie programmers, unless combined with experienced resources.

When to use the Agile Model

- Embrace Change
  When new changes are needed to be implemented. The freedom agile gives to change is very important. New changes can be implemented at very little cost because of the frequency of new increments that are produced. [Cer15]
- New Features
  To implement a new feature the developers need to lose only the work of a few days, or even only hours, to roll back and implement it. Changes can be discussed and features can be newly effected or removed based on feedback. [Cer15]
- Starting Quickly
  Unlike the waterfall model in agile model very limited planning is required to get started with the project. [Cer15]

When to use Agile Model

- Freedom of Options
  Having options gives them the ability to leave important decisions until more or better data or even entire hosting programs are available; meaning the project can continue to move forward without fear of reaching a sudden standstill. [Cer15]
- Less Documentation is Acceptable
  The agile methodology is not setting a lot of focus on design and documentation, these can't be critical in the project

Scrum

Figure: Taken from www.scrumalliance.org

Scrum

How do the Scrum Techniques and Methodologies Support CSE Development

Table: SCRUM [Blo12]

Product Owner		×
Team	(●)	×
Scrum Master	●	×
Sprints	● ⋆	
Backlogs	●	
Burndown Chart	●	
Sprint Planning Meeting	●	
Sprint Review	●	

less relevant: (no dot) | relevant: ● | extra relevant: ●⋆

Scrum

Interim Conclusion

Scrum is “on average” relevant and would work well for CSE development.[Blo12]

.

All points are judged as relevant, except product owner that is less relevant since external customers might not exist, and sprints that is extra relevant due to the uncertain nature of developing research software. If one lets the two outliers cancel each other out, **Scrum is ”on average” relevant and would work well for CSE development.**[Blo12]

A Detailed Look into Scrum

Product Owner

less relevant[Blo12]

Scrum

- The customer (or product owner) is a single point of interaction with the development team

CSE

- In CSE there might not be an external customer at all and this is hence deemed as less relevant

Product owner less relevant?!?

. . . One person should be the product owner that serves as the mediator between the customer organization and the Scrum team. . . . In projects without an external product owner, the team itself needs to act as the product owner . . . difficult in some cases.[Blo12]

A Detailed Look into Scrum

Product Owner

Pitfall

A common mistake is to have the product owner role filled by someone from the development team [Rot14] (or none at all!)

According to Johanna Rothman this is a mistake.

- ”When the business is unaccountable, the agile ecosystem breaks down.” [Rot14]
- ”It’s iterative and incremental, but it’s not even close to Scrum. It’s not agile.” [Rot14]

A Detailed Look into Scrum

Team

less relevant for lone researchers

Martin Blom:

"It is argued that Scrum works best for teams of 5-7 people, but most of the techniques can be used for single developers as well." [Blo12]

Scrum

- Recommended Size: 3-5-7 max 9

Working in teams require more than one person per definition. **For lone researchers this would hence be less relevant** than for typical SE projects that are normally done in groups, but **for larger CSE projects, teams would be as relevant as for non-CSE projects**.[Blo12]

A Detailed Look into Scrum

Scrum Master

relevant[Blo12]

Pitfall

Another common pitfall is for a scrum master to act as a contributor [Ber14]

While not prohibited by the Scrum methodology, the scrum master needs to ensure they have the capacity to **act in the role of scrum master first** and not working on tasks for the project.

Any team running Scrum would need a scrum master and CSE teams would be no different.[Blo12]

For more tips how to balance the roles visit:

Mission Possible: ScrumMaster and Technical Contributor, Page 3

A Detailed Look into Scrum

Sprints

extra relevant[Blo12]

from frame:
When to use Agile Model

- Embrace Change
- New Features
- Freedom of Options

Working in sprints, i.e. in iterations where a small part of the system is finished in each sprint, would suit a CSE developer well since **the exact specification of the system might not be available at first**. The more uncertainty there is, the more difficult it is to specify a complete requirements list directly.[Blo12]

XP

Figure: Taken from www.extremeprogramming.org

- XP is sometimes called "Scrum Compact"
- All the aspects of extreme programming are working together as a whole. Abandoning even one aspect might spoil the whole process

XP

Table: XP[Blo12]

Pair Programming	●	×
TDD	●	
Incremental Design	● ⋆	
Continuous Integration	●	
Collective Code Ownership	●	
Informative Workspace	●	
Coding Standard	●	
Sustainable Pace	●	

less relevant: (no dot) | relevant: ● | extra relevant: ●⋆

XP

Interim Conclusion

All key points of XP are relevant, except Incremental Development that was graded as extra relevant. XP would in other words suit CSE development well.[Blo12]

- In practical examples pair programming is seldom used see: [SHPL12]
- All the aspects of extreme programming are working together as a whole. Abandoning even one aspect might spoil the whole process.

Practical Examples

What Do We Know about Scientific Software Development's Agile Practices?[SHPL12]

- Web-Based Survey
- Literature Review
- Case Study Method

Practical Examples

Case Study

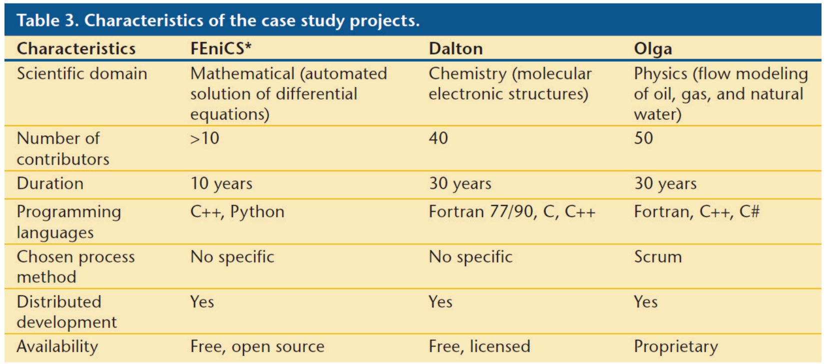

Table 3. Characteristics of the case study projects.

Characteristics	FEniCS*	Dalton	Olga
Scientific domain	Mathematical (automated solution of differential equations)	Chemistry (molecular electronic structures)	Physics (flow modeling of oil, gas, and natural water)
Number of contributors	>10	40	50
Duration	10 years	30 years	30 years
Programming languages	C++, Python	Fortran 77/90, C, C++	Fortran, C++, C#
Chosen process method	No specific	No specific	Scrum
Distributed development	Yes	Yes	Yes
Availability	Free, open source	Free, licensed	Proprietary

Figure: Characteristics of the case study projects

Practical Examples

Presence of Agile Practices

Interim Conclusion

Agile practices are indeed present in projects developing scientific software. . . . we couldn't find clear positive evidence as to their application.[SHPL12]

- 13 out of 35 agile practices are used in projects developing scientific software.[SHPL12]
- Practice 21 (all production code is pair programmed) had clear evidence that it's not used in most of the projects.[SHPL12]

Summary

Assessment of Agile Methods

Summary

Bewertung agiler Methoden

Nützt bei vielen Projekttypen und schadet zumindest nicht [Pec14]

- Do not rely on Agile solely as a process that will solve all the problems

Summary

Wrap-Up

- Under the right circumstances agile development suits CSE development very well
- Agile practices are not about "I" but about "we". It's about communication and feedback
- Agile practices are best used if you need to be flexible about change, new features and have freedom of options
- In practice not all agile methods are used even in projects dedicated to agile development
- People need to be experienced to use, adapt and understand the limits of agile methods

Summary

Avoiding Pitfalls

Tip

Avoiding Pitfalls

Before adopting agile methods for scientific development look up the common mistakes that have been made again and again in the last 20 Years. And make sure the project environment fits the agile principles.

Bibliography I

Steve Berczuk.
Mission possible: Scrummaster and technical contributor.
http://www.agileconnection.com/, 2014.

Martin Blom.
Is scrum and xp suitable for cse development?
Procedia Computer Science, 2012.
International Conference on Computational Science, ICCS 2010.

ISTQB Exam Certification.
What is agile model - advantages, disadvantages and when to use it?
http://istqbexamcertification.com/what-is-agile-model-advantages-disadvantages-and-when-to-use-it/, 2015.

Bruno Collet.
Limitations of agile software development.
http://www.brunocollet.com/blog, 2009.

Blue Star Infotech Ltd.
Distributed agile.
http://www.agiledistributed.com/distributedAgile.html, 2013.

Bibliography II

Juliana Medeiros, Alexandre Vasconcelos, and Carla Silva.
Integration of agile practices: An approach to improve the quality of software specifications, 2015.

Jörg Pechau.
Projektmanagement.
Department Informatik, Uni Hamburg, 2014.

Johanna Rothman.
When you have no product owner at all.
http://www.jrothman.com/, 2014.

Magnus Thorstein Sletholt, Jo Erskine Hannay, Dietmar Pfahl, and Hans Petter Langtangen.
What do we know about scientific software development's agile practices?
Computing in Science and Engineering, pages 24–36, 2012.

Dan Turk, Robert France, and Bernhard Rumpe.
Limitations of agile software processes.
Third International Conference on Extreme Programming and Flexible Processes in Software Engineering, pages 43–46, 2002.

Bibliography III

Wikipedia.
Wikipedia.
http://en.wikipedia.org.