Software Processes and Maintainance

Marta Kristín Lárusdóttir
School of Computer Science | Software Requirements and Design
1. November, 2019

Overview

- Software Processes
  - Waterfall
  - Scrum
  - Kanban
  - RUP
- Testing in Scrum Projects
- Maintainance of software

- Reading
  - ID book, 2.3.5

Additional reading material (optional)

- http://en.wikipedia.org/wiki/Waterfall_model
- http://en.wikipedia.org/wiki/Iterative_and_incremental_development
- http://en.wikipedia.org/wiki/Unified_process
- http://en.wikipedia.org/wiki/Rational_Unified_Process
- http://en.wikipedia.org/wiki/Scrum_(development)
- http://en.wikipedia.org/wiki/Kanban_(development)

A process versus a method

- A process implies a time-dependant order of steps/stages, and is a structured process of principles
- A method is a way to do something, not necessarily involving steps but can of course involve steps
- A set of methods can be used within a single process
- A distinction between the two often blurry
  - Many more methods are available than processes
  - the “think aloud” is a method not a process
- Sometimes software processes are called lifecycle models or frameworks
  - See 2.3.5 – pg. 64 in the book

What are the steps of the development process?

- The development process is at least 5 steps:
  - *Planning - Analysis – Design – Implementation – Testing*
    - Sometimes, "*Support, maintenance and improvement*" is added as a last step

- The precise mix of these steps depends on the methodology used
  - the Waterfall model, Scrum, Kanban or RUP

Deliverables - Documents

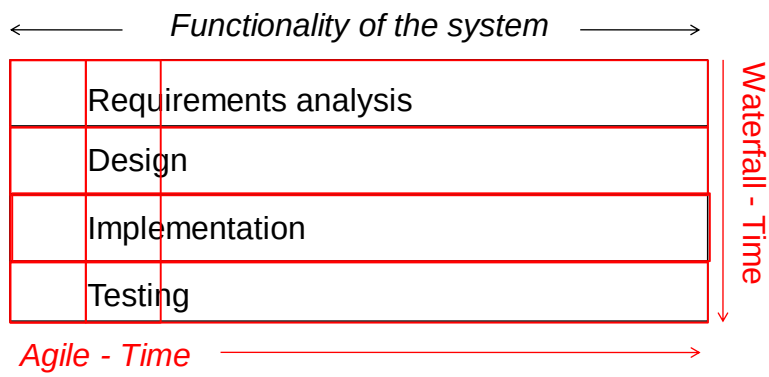
- Planning
  - Project proposal document
- Analysis
  - Requirement analysis report
- Design
  - System Design Documents
  - Test specifications
  - UI Design
- Implementation
  - Code
  - Documentation
- Testing
  - Test results, based on test spec

These Steps Need to be Taken

- Either all requirements analysed at the same time
  - Then you would do all the design, and so on !!
  - Waterfall process
- Or
  - Requirement analysis, design, implementation and testing in a 2 week period
- Either way
  - All these need to be done
  - It is just a matter of in which order

In Which Order?

- How big part of the whole system do you consider?
- What activities do you conduct?

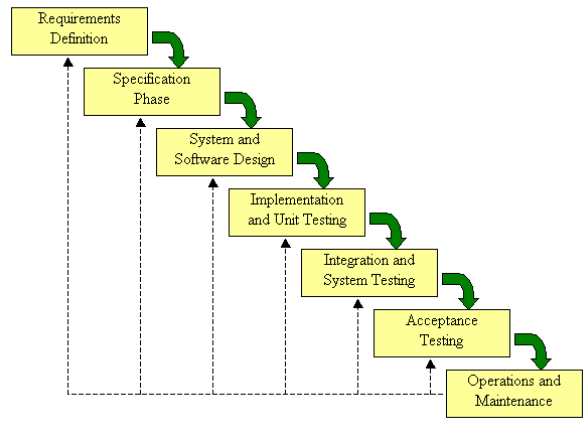


Focus of the course

- This course focuses on the analysis and design and some parts of testing
  - What it involves and methods which can be used
- We leave out
  - Project management
  - Big part of planning
  - Code development by programming languages (implementation)
  - The above will be learnt in other courses
- We started with requirement capture
  - In the analysis phase

The Waterfall Process

Waterfall Process

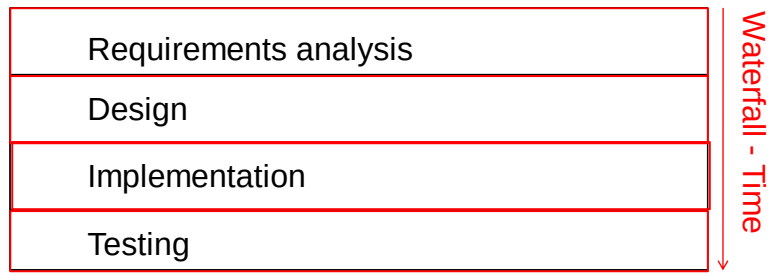


- Note: just like there are variations in what steps are specified in the development process, then we might see different versions of the waterfall model image

www.hr.is

The Waterfall Process

- The waterfall model specifies that we must finish each part of the development process for the whole system before we can move on to the next one
  - Has been criticized a lot



www.hr.is

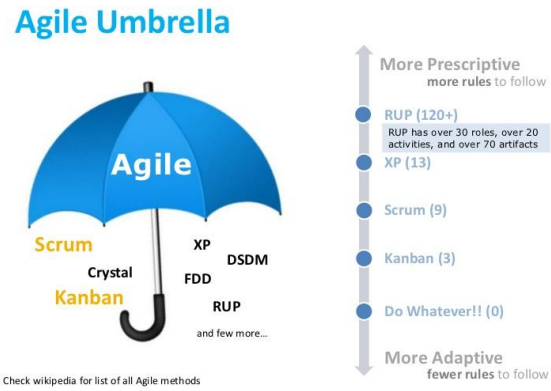
How would you build a bridge?

- Usually, we would first figure out what kind of bridge we need
  - what kind of traffic must go over the bridge
  - what kind of bridge would fit the landscape
- Then, the bridge would be designed
- It would be built
- ... and finally tested (does it handle the weight it is supposed to?)

Is writing software same as building bridges?

- No, there is a "soft" in software - we can often change things more easily in software development rather than when building a house/bridge/etc.
- However, this often means that people think that we don't need to design up front!
  - Which is absolutely not true!

Agile metods

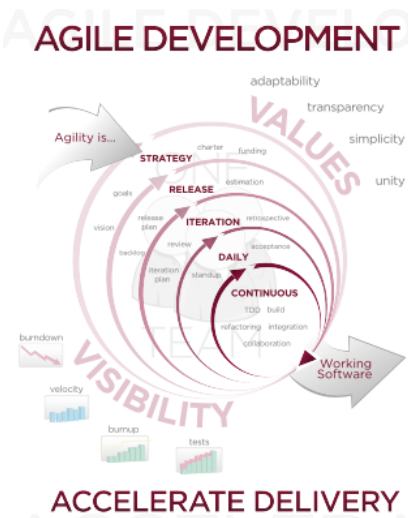


* Check wikipedia for list of all Agile methods

www.hr.is

The Agile (lean) methods

- Iterative and incremental development
  - As opposed to waterfall all the way
- Collaboration
- Adaptive
- Time-boxed
- Frequent delivery
- Customer satisfaction
  - User centered
- Examples of Agile methodologies
  - RUP, DSDM, Scrum, XP,



http://en.wikipedia.org/wiki/Agile_software_development

www.hr.is

Agile manifesto (2001)

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan
  - That is, while there is value in the items on the right, we value the items on the left more.[1]

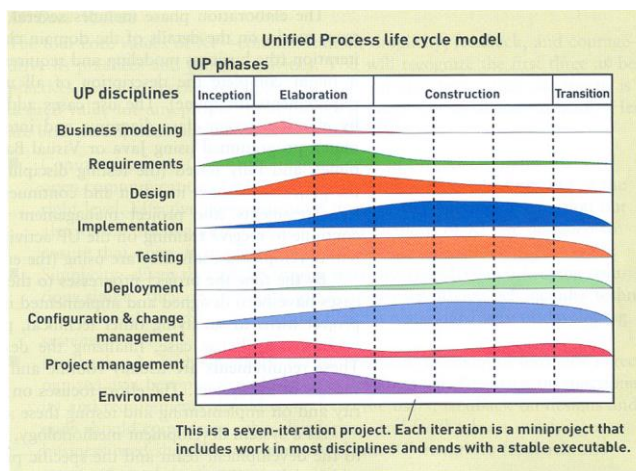
A short introduction to 3 methods

- These all are based on Agile principles, using various techniques and tools
- All based on adaptive and iterative approach to development
- The methodologies are (the book calls them methodologies):
  - UP (RUP)
  - Scrum
  - Kanban
- You will learn more about these in Hugbúnaðarfræði

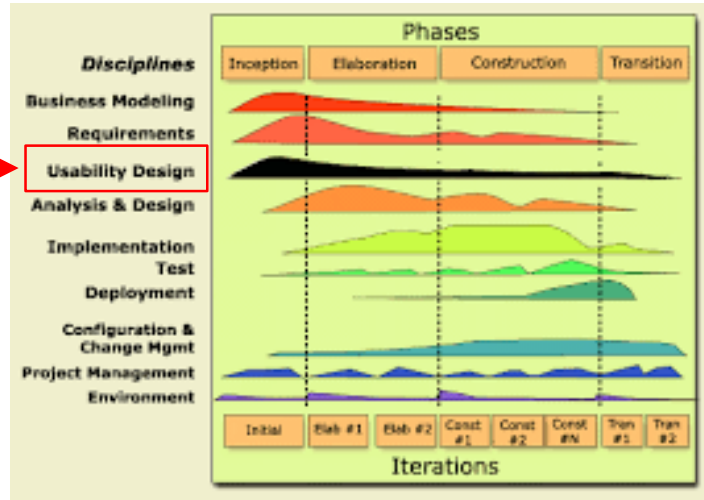
Unified Process (UP) methodology

- Developed by Rational Software (now part of IBM)
- Was very influential in OOP and adaptive methods
- New verions of the methodology are issued
- Uses UML as tool
- Has four phases:
  - inception, elaboration, construction, transition
- Has six disiplines
  - Business modeling, requirements, design, implementation, testing and deployment
- Has three additional support disciplines
  - Configuration and change management, Project management, and finally Environment

The UP life cycle



Adding Usability Design to RUP



Scrum

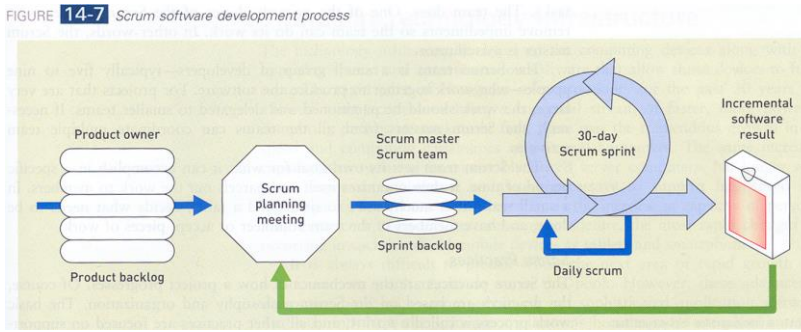
Scrum

- A concept from Rugby
  - Begins quickly, a very intensive effort, involves the entire team and usually only lasts for a short duration
- Product backlog
  - The basic control mechanism
  - Are the requirements (user functions, features and technology)
  - Are prioritized and only worked on a few items at a time
- Scrum organization
  - The Product Owner (owns and maintains the backlog)
  - The Scrum Master (a sort of project manager, a facilitator)
  - The Scrum Team or Teams (are self organized)

Scrum

- Very popular today

FIGURE 14–7 *Scrum software development process*



Scrum sprints

- Sprint (the basic work process)
  - A time controlled mini project that implements a specific portion of the system
  - Is time boxed (limited by time)
- A typical Sprint
  - A short planning session (usually <1day)
    - The sprint goal and prioritized deliverables from backlog
    - Work begins and scope is frozen
      - Changes go to the backlog list for following sprints
      - If too little time for planned deliverables, scope can be reduced
    - Daily Scrum: a short meeting every day
      - What have you done, what will you do, is anything stopping you?
    - Burn down: displays sprint progress of sprint backlog
    - Scrum Review at the end and a executable program

The main rules in Scrum (Cohn)

1. Work in iterations of no more than a month long
2. Be "done" with something
3. Get together and figure out what you're doing
4. Reflect on how well you did during the iteration
5. Talk a lot during the iteration

KANBAN

www.hr.is

Kanban

- Sometimes Scrum too big, then try Kanban
- Just-in-time delivery and Agile principles
  - Pull system: Tasks and processes displayed and developers pull from a queue
- The Kanban method does not prescribe a specific set of roles or process steps.
- Kanban starts with the roles and processes you already have and stimulates continuous, incremental and evolutionary changes to your system.
- Kanban encourages the use of scientific methods and leadership on all levels

www.hr.is

The 3 core practices of Kanban

1. Visualize (uses boards like Scrum)
2. Limit WIP (work in progress)
3. Measure the lead time/flow (monitor, measure, report)

- Make policies explicit (so everyone on the same page)
- Implement feedback loops (collaboration and review)
- Improve collaboratively, evolve experimentally (using models and the scientific method)

A few other principles from all over

- Finally, a few principles will be listed that are not a part of any one method/methodology
- You will run into them while reading material about the software development process, such as:
  - http://www.codinghorror.com/,
  - http://www.joelonsoftware.com/,
  - http://martinfowler.com/
  - and more

Software Development in Iceland

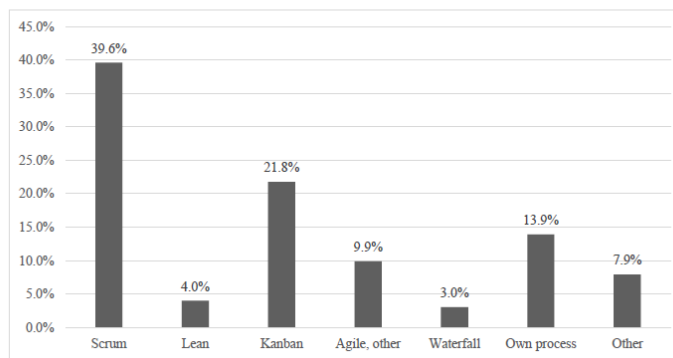
BS project in 2014 - Daniel Multykh

www.hr.is

Development Process

Figure 4 – Software development processes used for the last 3 months (N=73)

Question 12: What software development processes have you been using for the last 3 months? (You can select more than one option)

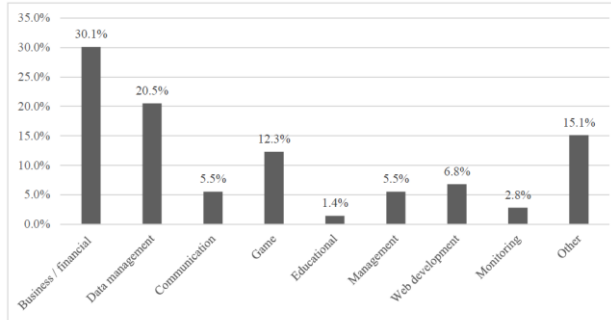


www.hr.is

Type of Software

Figure 2 – Type of software 1 (N=73)

Question 10: What type of software are you developing?



Main Job Roles

- Main Job roles
  - 86% programming,
  - 62% design,
  - 33% requirement gathering,
  - 33% testing
  - 65% in management
    - (Project, other) and concealing (teaching)

Table 1: Number of main job roles of the participants.

Main job role	Agile (N=32)		Other (N=10)	
One	6	19%	1	10%
Two	8	25%	3	30%
Three	10	31%	3	30%
Four	5	16%	2	20%
Five	3	9%	1	10%

The Focus on Usability Testing in Scrum Projects in Iceland

Paper presented at the World Computer Congress in Australin 2010

www.hr.is

The Authors

Marta Lárusdóttir

Emma Rún Bjarnadóttir

Jan Gulliksen

HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

Calidris

KTH
VETENSKAP
OCH KONST

www.hr.is

The Motivation for the Study

- Scrum has become popular in resent year in Iceland
  - 1/3 of respondent of a survey for software developers in Iceland used Scrum in spring 2009
  - The popularity is growing
- The motivation
  - What testing is conducted?
  - How often is it conducted?
  - How easy is it to do testing in Scrum projects?
  - What are the hindrances of conducting testing?
  - What is the difference of usability and accessability testing?

The Study

- A survey to gather information on
  - How testing is conducted
  - To what extent testing techniques are used
- Interviews were conducted
  - To exemplify how practitioners conduct usability testing
  - How they describe usability testing compared to acceptance testing
- The research questions:
  1. How is testing practiced in Scrum projects in the industry?
  2. To what extent is usability testing performed compared to other testing techniques?
  3. How does usability testing differ from acceptance testing in Scrum projects?

The Research Questions

1. How is testing practiced in Scrum projects in the industry?
2. To what extent is usability testing performed compared to other testing techniques?
3. How does usability testing differ from acceptance testing in Scrum projects?

Research Methods

- Questionnaire
  - 26 questions, 21 multiple choice and 5 open
    - Background and experience of the respondent
    - The company where the respondent works
    - The software development process(es) used
    - To which extent and who is conducting different testing techniques
    - The change in conducting software testing when compared to previous/parallel software development process
  - Was sent out to 20 companies using the Scrum process
    - 25 responds from 18 companies
- Interviews
  - 6 persons, 3 software testers and 3 Scrum Masters
  - Main focus on usability and acceptance testing

The Respondents in the Survey

- 25 respondents from 18 companies
  - 76% have a degree in computer science or engineering
  - 68% male, 20% female, 12% did not reply
- Had various experience in the software industry
  - 16% > 15 years, around 25% 10 – 15 years, 4 – 9 years and 1- 3 years
- Scrum used in all the companies
  - Were asked to estimate the extent to which they use Scrum
    - Almost half 81 – 100%, 34% say 21 – 80%, 22% say 0 – 20%
  - 44% said that they use their own process beside Scrum
  - The size of the companies was various
    - 33% up to 19 employees, 28% 20 – 59 emp. , 33% over 60 employees
- Roles
  - 44% Scrum Masters, 24% testers, 20% Product Owners, 12% other

Scrum Activities Used

Scrum activity	Percent of respondents
The role of Product Owner is used	88%
The role of Scrum Master is used	92%
Requirements/user stories are kept in a Product backlog	84%
Planning poker is played on Product backlog items	60%
A Sprint backlog is established at the beginning of a sprint	92%
Each sprint/iteration is 2 - 4 weeks long	96%
Potentially shippable product exists at the end of a sprint	64%
A sprint review meeting is held at the end of a sprint	80%
A sprint retrospective meeting is held at the end of a sprint	88%
Scrum (Agile) metrics are used, like burn down charts	76%

The main rules in Scrum (Cohn)

1. Work in iterations of no more than a month long
   - Mainly 2 – 4 weeks long (96%)
2. Be "done" with something
   - Potentially shippable product (64%)
3. Get together and figure out what you're doing
   - Sprint backlog made (92%)
4. Reflect on how well you did during the iteration
   - Retrospective meeting (88%)
5. Talk a lot during the iteration
   - Don't have a measure for that!

Testing Practices in Scrum

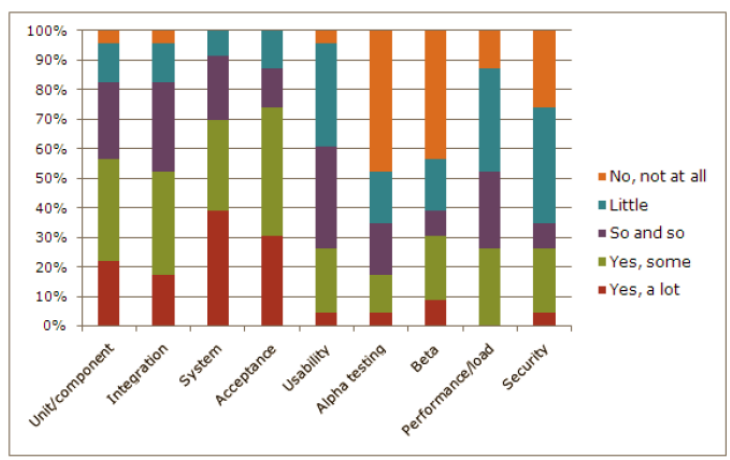
Testing practice	Percent of repondents
Software testing falls within the frame of "done" in each sprint	64%
Software testing is squeezed into the end of each iteration	36%
Software testing is not well integrated with coding and ends up one sprint behind	20%
Software testing is performed in a separate test environment	44%
Good management of version control	60%
Before a major version release, there is a bug-fix sprint	40%
Software testing became easier than in a parallel/prior process	44%
Overall more software testing is done than in a parallel/prior process	44%
Overall less software testing is done than in a parallel/prior process	12%
Programmers started using more test-driven development/design	48%
Software testers became more involved throughout the whole development	72%

Description of Testing Techniques

Testing technique	Description
Unit/component testing	The testing of individual software components.
Integration testing	Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems.
System testing	The process of testing an integrated system to verify that it meets specified requirements. This includes test design techniques like boundary valued analysis and is usually done by internal software testers.
Acceptance testing	Formal testing with respect to user needs, requirements and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the users, customers or other authorized entity to determine whether or not to accept the system.
Usability testing	Testing to determine the extent to which the software product is understood, easy to learn, easy to operate and attractive to the users under specified conditions.
Alpha testing	Simulated or actual operational testing by potential users/customers or an independent test team at the developer' site, but outside the development organization. Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing.
Beta testing	Operational testing by potential and/or existing users/customers at an external site not otherwise involved with the developers, to determine whether or not a component or system satisfies the user/customer needs and fits within the business processes. Beta testing is often employed as a form of external acceptance testing for off-the-shelf software in order to acquire feedback from the market.
Performance/load testing	The process of testing to determine the performance and/or measuring the behavior of a component or system with increasing load, e.g. the number of parallel users and/or numbers of transactions, to determine what load can be handled by the component or system.
Security testing	Testing to determine the security of the software product.

www.hr.is

To Which Extent is Testing Done?



To which extent are the following types of software testing done in Scrum (Agile) projects? Each type may be done by your Scrum (Agile) team members (company internal) and/or customers (company external), see next question.

www.hr.is

Using a Testing Techniques Less than Others

Testing technique	Lack of training/ knowledge	Lack of budget	Lack of time	Other	N/A	N
Unit/compontent testing	**36%**	0%	32%	5%	27%	22
Integration testing	11%	0%	42%	0%	**47%**	19
System testing	7%	0%	**47%**	0%	**47%**	15
Acceptance testing	7%	0%	27%	7%	**60%**	15
Usability testing	*20%*	*15%*	***35%***	*10%*	*20%*	*20*
Alpha testing	0%	11%	11%	10%	**68%**	19
Beta testing	0%	11%	17%	11%	**61%**	18
Performance/load testing	26%	11%	**32%**	0%	**32%**	19
Security testing	**47%**	5%	16%	0%	32%	19

The importance of Usability Testing

- Usability testing is important
  - None wanted to ignore it – wanted more testing occasionally
- Testers are often part of the teams
  - Know what functionality they are going to test
  - Test simultaneously
- BUT did not have time for usability testing
  - “Could be good to do it once a year”
  - “It is always on my mind”
  - The increments are not that big, no need for usability testing
  - The users are sometimes not willing to take part

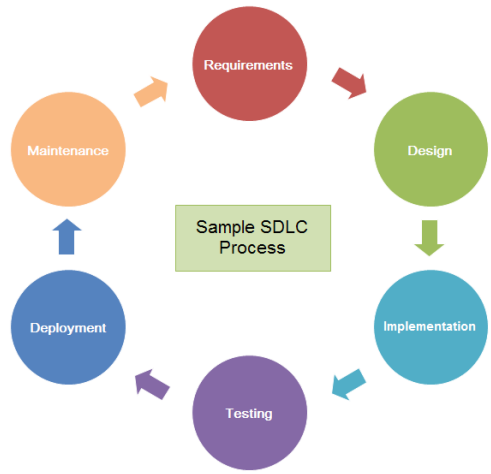
Difference of Acceptance and Usability testing

- Acceptance testing more structured
  - The customer has to sign that he or she has accepted
- Usability testing
  - User are often involved in that testing
  - Sometimes done by external usability specialist
- User acceptance – what is that?
  - Not a formal usability testing procedure
  - The users are asked to go through predefined steps
  - Asked to report usability problems
  - Done at the end of a project

Summary from the study

- The fundamental roles, activities and artifact in Scrum
  - are used to great extent
- Testing became easier in Scrum
  - Than in prior process – only a few said less testing was done
- Usability and performance testing are similar
  - Unit, integration, system and acceptance testing are much more frequent
  - The testers want to carefully plan their tests
  - They do not have time for that in Scrum
- The implication for further work
  - Find ways of testing the usability on a smaller scale

Maintainance Phase



www.hr.is

Let's be realistic

- Nowadays we are not often making completely new systems
- No matter where you will go on working, there will probably always be work to be done on already existing systems
- And it can be just as fun ☺

www.hr.is

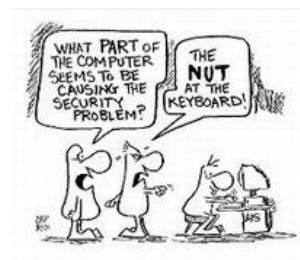
A maintenance phase

- We need to be able to change software so that business opportunities will not be lost
- Maintenance phase is usually much longer than the initial making of the original system
- The maintenance phase also consumes a large part of the overall cost of the system's lifecycle
- This is often forgotten by the buyers of the system, so be aware of that

http://search.dilbert.com/comic/Maintenance

The maintenance phase, what is involved?

- After a system has been installed, three major activities occur which make up the maintenance phase:
  - Maintaining the system
  - Enhancing the system
  - Supporting the users
    - (see next 3 slides)



http://www.piculous.com/jokesfunny-computer-tech-support/

Maintaining the system

- Fixing errors
  - e.g. all kinds of bugs or faulty design decisions
- Making minor adjustments
  - e.g. new attribute in a database table, new field in a form, adjusting calculations, etc...

"Here's your problem. The software was manufactured in November and your computer was manufactured in February. Sagittarius is incompatible with Aquarius."

http://www.cartoonstock.com/directory/c/computer_maintenance.asp

Enhancing the system

- All major modifications
  - e.g. new business requirements, new system infrastructure, new law calls for new data to be available, new technology, to name but a few.
- Almost always calls for a new version of the system

http://search.dilbert.com/search?p=Q&srid=S3USCDR02&lbc=dilbert&ts=custom&w=tech%20support&uid=805574756&method=and&isort=date&view=list&filter=type%3acomic&srt=6

Supporting the users

- Help desk
  - Common in USA to put newly hires there
  - Not always well set up, nor available
- Training new users
  - Often overlooked, but is often a major part in the system's success
  - New people may start using the system long after you installed it
- Keeping all documentation up-to-date
  - Very difficult and takes a lot of time and effort
  - Very often so badly done
- All of the above are a continuous battle

http://jccsst-random.blogspot.com/2012/10/tech-support.html

What about analysis and design?

- But how do we go about analysis and design on already existing systems. The short answer is:
  - It is the same process as when developing a new system (especially true for the major enhancements)
    - We need to go through all of that we have learnt during this course, e.g. gather & prioritize the requirements, know and talk to users, prototype and iterate, test and test and then test more, make code design decisions, etc…

And some more

- Very important not to forget the user during the maintenance phase, even more critical than before
- And to prototype how a new/changed feature should best fit into the already existing system
- Also there should be a known path of how users can request modifications or report problems which can result in Change requests (beiðni)
- Do not just change a system to change it, make justifiable decisions with stakeholders on what should be changed and what not

Summary

- Different processes for software development
  - Waterfall
  - Scrum, Kanban, RUP
- Testing on Icelandic software industry
- Maintenance phase