Introduction to Scrum

September 1, 2011

David Broman
Department of Computer and Information Science
Linköping University, Sweden
davbr@ida.liu.se

The Waterfall model

- One of the first life-cycle models (Royce, 1970)
- Very common, very criticized

Why is the waterfall model so criticized?
Which are the problems?
Can it be useful sometimes?

- Finish each phase before continue to next.
- Milestone and deliverable at each step. (Artifacts such as Design document, Req. Specification. etc.).

Part I
Processes and models in Software Engineering

Part II
SCRUM
The Waterfall model - some arguments

**Pros**
- Simple, manageable and easy to understand
- Fits to common project management practices (milestones, deliverables etc.)
- Focus on requirements and design at beginning, save money and time at the end
- Can be suitable for short projects (some weeks)
- Can be suitable for "stable" projects, where requirements do not change
- Focus on documents, saves knowledge which can be reused by other people.
- Widely used, e.g. US Department of Defense
- Can be suitable for fixed-price contracts

**Cons**
- Software requirements change, hard to sign-off on a SRS.
- Early commitment. Changes at the end, large impact.
- Feedback is needed to understand a phase. E.g. implementation is needed to understand some design.
- Difficult to estimate time and cost for the phases.
- Handling risks are not part of the model. Pushes the risks forward.
- Software "is not" developed in such a way. It evolves when problems are more understood. Little room for problem solving.
Can we improve the model?

Part I
Processes and models in Software Engineering

Part II
SCRUM

David Broman
davbr@ida.liu.se

Do it twice?

Part I
Processes and models in Software Engineering

Part II
SCRUM

David Broman
davbr@ida.liu.se

The original paper is actually misunderstood!
(Royce, 1970) includes
- Iteration of phases
- "Do it twice" prototype
Is overlapping phases a solution?

When do we "sign-off", e.g. when do we have all requirements?

What if a major design flaw is discovered at the testing phase?

Time

requirements

design

implementation

test

Release!

What kind of structure have we actually achieved? No sign-off. How does this help us?

What if a major design flaw is discovered at the testing phase?
"The hardest single part of building a software system is deciding precisely what to build"  
(Frederick P. Brooks)

When should the releases take place?  
**Time-boxing** - The time period is fixed for each iteration.

What should be included in the release?  
**Prioritized functionality** - Do the most important parts first.

Iterative Development

When should the releases take place?  
**Time-boxing** - The time period is fixed for each iteration.

What should be included in the release?  
**Prioritized functionality** - Do the most important parts first.
We are using an iterative process!

Define a plan with 1..N iterations. We do not have to care about plans...

Now, let's hack!

Is this a good iterative process? Of course not. We need some structure!

Methodologies and defined Process frameworks

Agile Approaches - Agile Alliance

Lightweight approaches to satisfy the customers with "early and continuous delivery of valuable software"

Manifesto for Agile Software development

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

(http://agilemanifesto.org, 2001)
Approach public in 1995 at OOPSLA

"Scrum" strategy used in rugby for getting an out-of-play ball back into play.
Scrum Overview

Roles
- Team
- Product Owner
- Scrum master

Lists
- Product backlog
- Sprint backlog
- Impediment list

Meetings
- Release planning
- Sprint planning
- Daily Scrum
- Sprint review
- Sprint retrospective

Sprint, Task board, Burn-down chart, Done, Velocity

The Sprint (1)

Roles
- Team
- Product Owner
- Scrum master

Lists
- Product backlog
- Sprint backlog
- Impediment list

Meetings
- Release planning
- Sprint planning
- Daily Scrum
- Sprint review
- Sprint retrospective

An iteration
- Time-boxed
- 30 days or less
- No time between sprints

- 40 hours week
- Open and visible

Sprint, Task board, Burn-down chart, Done, Velocity
The Team

Roles
- Team
- Product Owner
- Scrum master

Lists
- Product backlog
- Sprint backlog
- Impediment list

Meetings
- Release planning
- Sprint planning
- Daily Scrum
- Sprint review
- Sprint retrospective

Sprint, Task board, Burn-down chart, Done, Velocity

Product Owner

Roles
- Team
- Product Owner
- Scrum master

Lists
- Product backlog
- Sprint backlog
- Impediment list

Meetings
- Release planning
- Sprint planning
- Daily Scrum
- Sprint review
- Sprint retrospective

Sprint, Task board, Burn-down chart, Done, Velocity

- Cross functional
- No titles
- Self-organized
- 7 plus/minus two
- Develops, tests, documents etc. in iterations = sprints

Product Owner

- One and only one person
- Prioritize and manage the product backlog
- Manage ROI
- The customer "interface"

The product owner may not:
- act as a project manager
- tell when and what something should be done
Scrum Master

Roles
• Team
• Product Owner
• Scrum master

Lists
• Product backlog
• Sprint backlog
• Impediment list

Meetings
• Release planning
• Sprint planning
• Daily Scrum
• Sprint review
• Sprint retrospective

• Make sure the scrum team adheres to Scrum values, practices and rules
• Run meetings
• Protects the team from disturbance
• Collects and removes obstacles (Impediment list)

The scrum master may not
• Manage the scrum team - the scrum team is self-organized

Scrum master cannot be product owner

Sprint, Task board, Burn-down chart, Done, Velocity

Pigs and Chickens

Roles
• Team
• Product Owner
• Scrum master

Lists
• Product backlog
• Sprint backlog
• Impediment list

Meetings
• Release planning
• Sprint planning
• Daily Scrum
• Sprint review
• Sprint retrospective

• Scrum team members are "pigs"
• Everyone else is a "chicken"
• Chickens cannot tell "pigs" how to do their work

"A chicken and a pig are together when the chicken says "Let's start a restaurant!". The pig thinks it over and says "What would we call this restaurant?" The chicken says "Ham n' Eggs!" The pig says "No thanks, I'd be committed, but you would only be involved!"
Exercise
Project leading and selforganization

Product Backlog (2)

- List of product backlog items (PBI)
  (approx. List of potential requirements)
- Prioritized
- Available
- Never complete
- Features, bug fixes, documentation, tests etc.
- Value (PO) and estimates (Team)
- PO responsible, everyone can add

Roles
- Team
- Product Owner
- Scrum master

Lists
- Product backlog
- Sprint backlog
- Impediment list

Meetings
- Release planning
- Sprint planning
- Daily Scrum
- Sprint review
- Sprint retrospective
Part I – "What" (4)
- Break down top items
- Estimate product backlog
- Select PBIs for a sprint
- Time-boxed 4h

Optional: Sprint GOAL
Prioritization of requirements

Part I
Processes and models in Software Engineering

David Broman
davbr@ida.liu.se

Part II
SCRUM

Effort Estimation… a good approach?

Part I
Processes and models in Software Engineering

David Broman
davbr@ida.liu.se

How long time does it take for you to implement the encryption layer?

Sam
the seller

Harry
the hacker

No idea. I have never done this before... I wonder if it is even possible.

8 months +- 2 months
**Planning Poker® - Exercise**

**Customer value estimation**
- Talk to the customer!

**Effort estimation (Planning poker®)**
- Variant of the Delphi approach
- Speed-up estimations
- Everyone participate

1. List things to be done to make my party great!
2. Prioritize the list in regards to customer value
3. Base line e.g., “Cook food” is 2. Relative estimation.
   Choices: ? 0 ½ 1 2 3 5 8 13 20
   40 100 infinity
4. Effort estimation for each item
   - Read and discuss item (short)
   - Make a decision (privately)
   - Show cards at the same time
   - Highest and lowest argue
   - Repeat until consensus

**Part I**
Processes and models in Software Engineering

**Part II**
SCRUM

---

**Sprint planning, Part 2, ”how”**

**Sprint Backlog(6)**

**Roles**
- Team
- Product Owner
- Scrum master

**Lists**
- Product backlog
- Sprint backlog
- Impediment list

**Meetings**
- Release planning
- Sprint planning
- Daily Scrum
- Sprint review
- Sprint retrospective

**Part 2 - ”How” (5)**
- Design
- Identify tasks
- Estimate tasks (hours)
- Output: **Sprint backlog**

**Sprint backlog**
- Consists of tasks
- Less than 2 day (preferable hours)
- Not ordered

**Sprint, Task board, Burn-down chart, Done, Velocity**
Task Board

- Not defined by Scrum
- Use non high-tech
- Example of columns
  - PBI
  - Todo
  - In Process
  - To Verify
  - Done

Roles
- Team
- Product Owner
- Scrum master

Lists
- Product backlog
- Sprint backlog
- Impediment list

Meetings
- Release planning
- Sprint planning
- Daily Scrum
- Sprint review
- Sprint retrospective

Sample Task Board
Part II

SCRUM

• When are we done?
• “No more remaining work”
• Includes testing, documentation etc.
• Possible to ship after each sprint
• Everybody – understand what done means

Tools to support done
• Version handling (SCM)
• Automated build
• Automated tests (Continuous integration)

Roles
• Team
• Product Owner
• Scrum master

Lists
• Product backlog
• Sprint backlog
• Impediment list

Meetings
• Release planning
• Sprint planning
• Daily Scrum
• Sprint review
• Sprint retrospective

Sprint, Task board, Burn-down chart, Done, Velocity

Part I
Processes and models in Software Engineering

Part II
SCRUM

Burn-down chart
• Only track hours remaining, not hours worked
• X – days (in Sprint)
• Y – hours remaining
• Remove meeting time, vacation etc. from total available hours
• Update only when PBIs are DONE
• Slope – the team velocity.

• When not done – Undone PBIs

Sprint, Task board, Burn-down chart, Done, Velocity
Daily Scrum (8)

Part I
Processes and models in Software Engineering

Part II
SCRUM

Roles
• Team
• Product Owner
• Scrum master

Lists
• Product backlog
• Sprint backlog
• Impediment list

Meetings
• Release planning
• Sprint planning
• Daily Scrum
• Sprint review
• Sprint retrospective

• Stand-up meeting
• Scrum master runs the meeting
• Every morning
• Time-boxed 15min
• 1 minute each person
• Do NOT solve problems!
• Chickens are not allowed to talk

Questions
• What did you do yesterday?
• What will you do today?
• What impediments stand in your way?

Sprint, Task board, Burn-down chart, Done, Velocity

Exercise
Daily Scrum - Theater
Impediment List

Roles
- Team
- Product Owner
- Scrum master

Lists
- Product backlog
- Sprint backlog
- Impediment list

Meetings
- Release planning
- Sprint planning
- Daily Scrum
- Sprint review
- Sprint retrospective

- List of obstacles
- Scrum Master’s backlog
- Daily update
- Open, visible and honest

Sprint, Task board, Burn-down chart, Done, Velocity

Sprint review (9)

Roles
- Team
- Product Owner
- Scrum master

Lists
- Product backlog
- Sprint backlog
- Impediment list

Meetings
- Release planning
- Sprint planning
- Daily Scrum
- Sprint review
- Sprint retrospective

- Time-boxed 4h
- End of sprint
- Team + Scrum Master + Product owner (potentially also customer)

- Informal meeting – what has been done
- Demonstrate – no power points
- Show working functionality – get feedback

Sprint, Task board, Burn-down chart, Done, Velocity
Sprint retrospective (10)

Roles
• Team
• Product Owner
• Scrum master

Lists
• Product backlog
• Sprint backlog
• Impediment list

Meetings
• Release planning
• Sprint planning
• Daily Scrum
• Sprint review
• Sprint retrospective

• 3h, time-boxed
• Inspect the last sprint, regarding
  • People
  • Relationships
  • Processes
  • Tools
• How to make things better
  – process improvements

• What was good?
• What was not so good?
• How can we improve?

Roles
• Team
• Product Owner
• Scrum master

Lists
• Product backlog
• Sprint backlog
• Impediment list

Meetings
• Release planning
• Sprint planning
• Daily Scrum
• Sprint review
• Sprint retrospective

Thanks for listening!

Sprint, Task board, Burn-down chart, Done, Velocity