

Requirements Engineering and SCRUM

Peter Dolog
dolog [at] cs [dot] aau [dot] dk
E2-201
Information Systems
February 13, 2007

Scrum

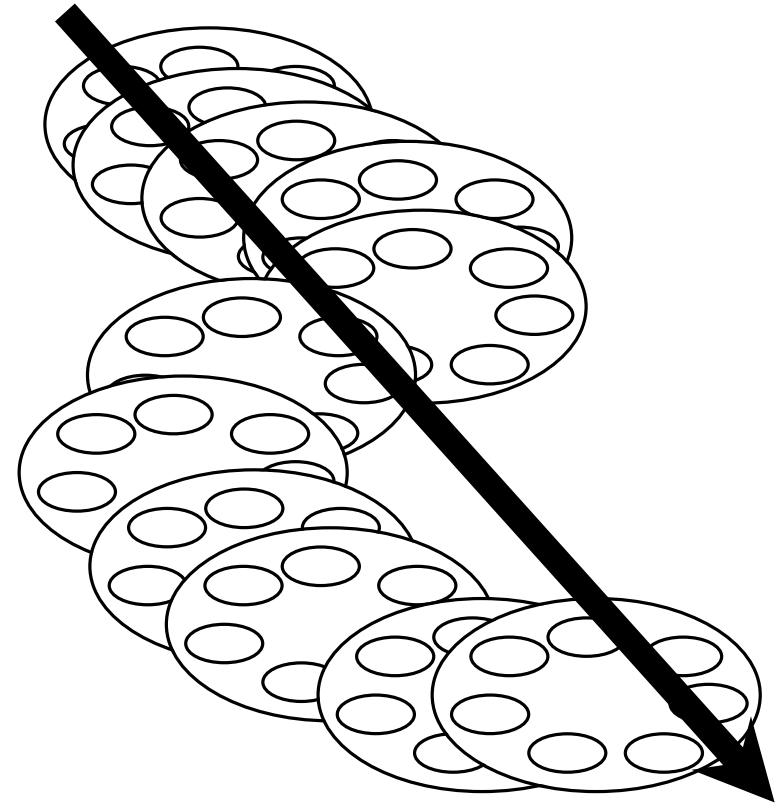
Larman Ch. 7

Scrum Model



A small group is responsible for picking up the ball and moving it toward the goal.

Start



Goal

SCRUM Practices

Product Backlog

Sprint

Sprint Planning Meeting

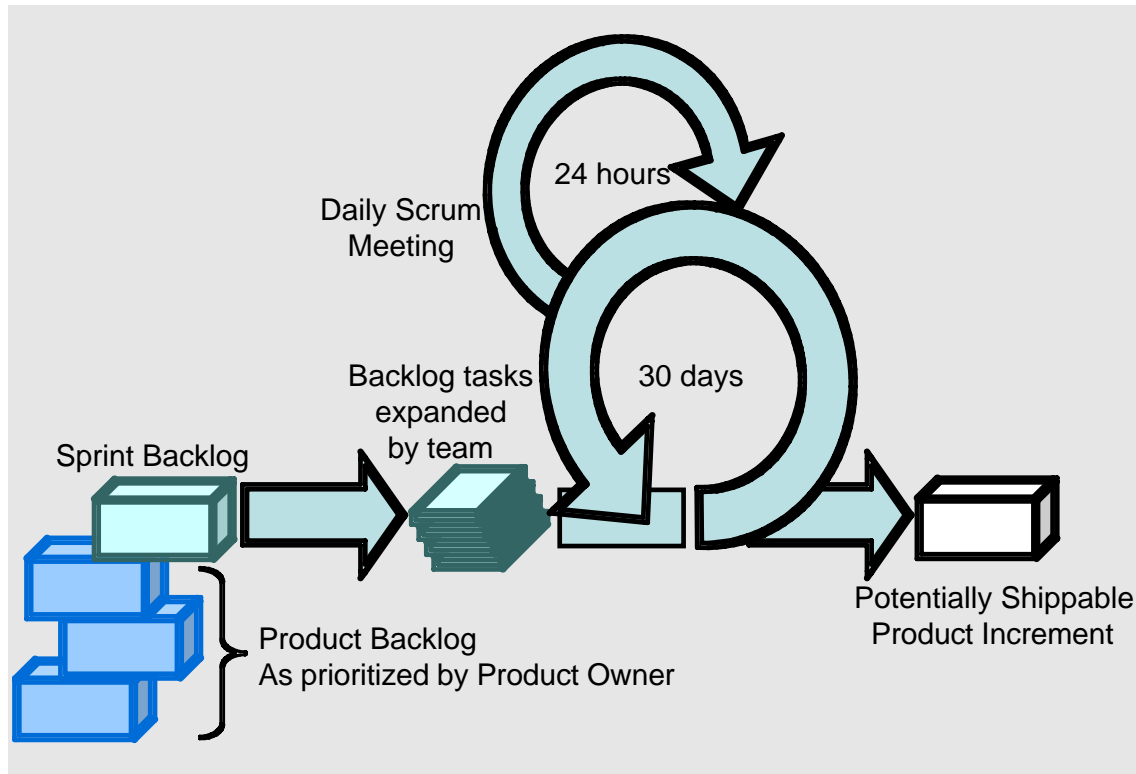
Sprint Backlog

Roles - Product Owner, Scrum Master, Scrum Team

Daily Scrum Meeting

Sprint Review Meeting

Scrum Lifecycle



Product Backlog

Prioritized list of work to be performed on a product
Anyone can contribute backlog items
Product Owner responsible for prioritisation

Sprint

A fixed period of 30 days to develop a deliverable product

The Sprint includes design, coding, testing, and documentation

Once a Sprint has started only the Scrum Team can add or remove items in Sprint backlog

Abnormal termination of Sprint is called for when the Sprint Goal no longer makes sense



Sprint Planning

Meeting to set the next Sprint goal

Product
Backlog

Team
Capabilities

Business
Conditions

Technology
Stability

Executable
Product
Increment

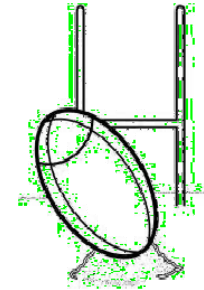


Review
Consider
Organize

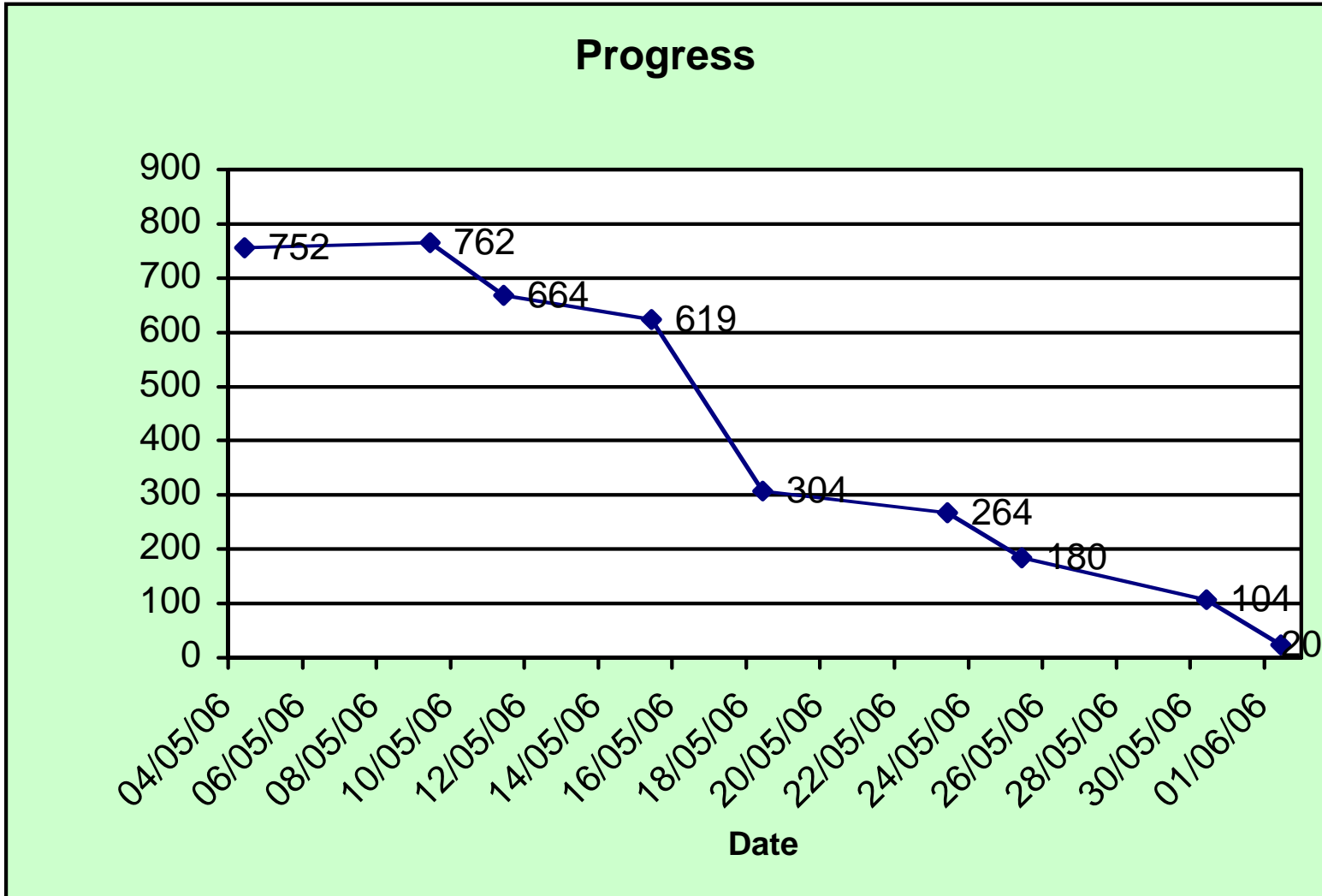


Next Sprint
Goal

Sprint Backlog



Sprint Burndown Chart



Daily Scrum

Daily 15 minute status meeting

Team stands in a circle facing each other

Each team member answers 3 questions:

- What have you done since the last Scrum?
- What will you do between now and the next Scrum?
- What got in your way of doing work?

For synchronization not problem solving!



Sprint Review

During this meeting the team presents to management, customers, users and the Product Owner the product increment that has been built during the Sprint

The team tells the story of its journey during the Sprint.

Powerpoint presentations are forbidden!



Scrum Team

Self-organizing

Cross-functional with no roles

Seven plus or minus two

Responsible for committing to work

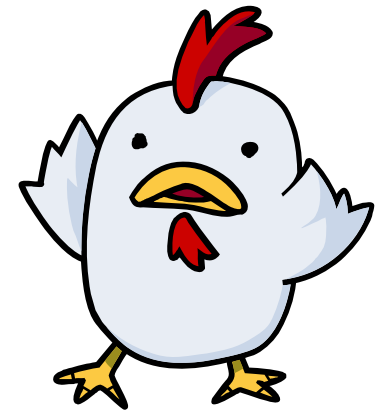
Authority to do whatever is needed to meet commitment



Chickens & Pigs

Members of Scrum Team are known as Pigs because they are committed to delivering Sprint Goal

People who are involved but not dedicated to the project are known as Chickens - they attend Scrum meetings as observers



Product Owner

Sets development schedule by prioritizing backlog

One person in this role ensures that only one set of requirements drives development

Can be influenced by committees, management, customers, sales people, but is the only person that prioritizes

Works with others to estimate items on Product Backlog

Eliminates confusion of multiple bosses, different opinions, and interference



Scrum Master

The Scrum Master is responsible for

- the success of SCRUM
- establishing SCRUM practices and rules, shielding the team and removing obstacles

representing management to the project



How Big is a Team?

Typically 5-10 people

Mike Cohn has led teams of 100+

Ken Schwaber has led teams of 600+

Obviously, very large teams are a very special case

“Scrum of Scrums” technique

Software Requirements

Bashar Nuseibeh and Steve Easterbrook

Requirements Engineering

”Requirements engineering is the branch of software engineering concerned with the real-world goals for, functions of, and constraints on software systems. It **is** also concerned with the relationship of **these** factors to precise specifications of software behavior, and to their evolution over time and across software **families.**”

Software Requirements Processes

eliciting requirements,
modelling and *analysing* requirements,
communicating requirements,
agreeing requirements, and
evolving requirements.

Context and GroundWork

Specific customer

Market driven

Different techniques for different type of the software

Preparation

Assessment on feasibility

High level conflict assessment

Identification of a suitable process for RE with suitable method
and techniques

Requirements Elicitation

Elicitation and capturing

Boundaries

Stakeholders

Goals

Tasks

Use Cases

Elicitation Techniques

Data access: questionnaires, surveys, interviews, documentation,

...

Group elicitation techniques: workshops, brainstormings

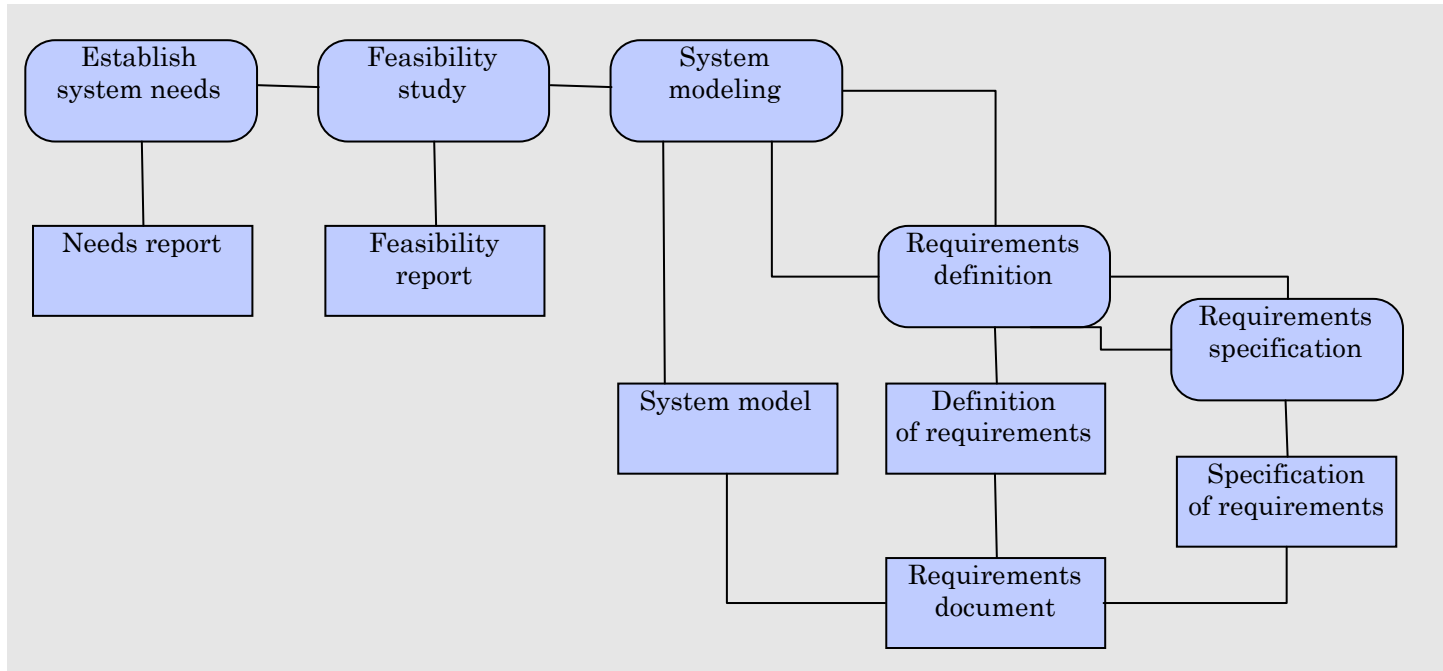
Prototyping

Model-driven techniques such as goal oriented or scenario oriented

Cognitive techniques: protocol analysis, laddering, card sorting, repository grids

Contextual techniques: participant observation

Requirements Formulation



Communicating requirements

To communicate to different stakeholders

Specification languages

Requirements management: traceable and readable by the others

Standards to document and structure

Traceability: read, navigate, query and change

Agreeing the Requirements

Validation

Inspection and formal analysis: coherence

Prototyping and specification animation and scenarios:
correspondence to real world problems

What we know?

Social agreement?

Evolving Requirements

Version control and configuration amangement

Impact of req. Change on other products of SEProcess

Consistency

Viewpoints framework

Software Product Lines

Applications sharing common requirements and architecture

Yet differ in some key requirements

Feature Models: common and variable features and dependencies