

COMP 3111

SOFTWARE ENGINEERING

TUTORIAL:

PROJECT MANAGEMENT USING SCRUM

PROJECT MANAGEMENT

Managing software development projects is not easy!

- It requires:
 - **good organization** → **divide** and **conquer** approach
 - **divide work** into smaller pieces (tasks)
 - **assign appropriate people** (or teams) to tasks
 - **good communication** → both **formal** and **informal**
 - regular meetings
 - tracking progress

**Poor project organization and/or communication
can be *fatal* to a software development project!**

PROJECT ORGANIZATION: COMPONENTS

A software development project consists of four inter-related parts.

- **Participants** → People who do the work.
 - A person participating in the project.
- **Tasks** → Things that need to be done.
 - The work to be performed by a participant.
- **Work Products** → Things that get produced.
 - An item (artifact) produced by a task (e.g., models, code, etc.—some are *deliverables*).
- **Schedule** → Who does what and in what order.
 - Tasks need to be prioritized, ordered and assigned to participants.
 - Some tasks can be done in parallel, others must be sequenced.

PROJECT ORGANIZATION: PARTICIPANTS

- Participants can work as **one big team** or be organized into **smaller project teams** → **there may be many project teams**.
- An **individual** or a **project team** **carries out** some **well-defined task** in the project.
- **Each participant plays a certain role** (e.g., management, development, cross-functional (liaison), consultant).
- **Management roles** (see tutorial notes for duties and responsibilities):
 - **project manager**: manages a project.
 - **team leader/person in charge (PIC)**: manages a project team.

PROJECT COMMUNICATION

- **Planned Communication**
 - Problem inspection
 - Client review
 - Project review
 - Peer review
 - **Status review** → The focus of COMP 3111 meetings!
 - Brainstorming
 - Releases
 - Postmortem review
- **Unplanned Communication**
 - Request for clarification
 - Request for change
 - Issue resolution



PROJECT MANAGEMENT USING SCRUM

- Scrum is an agile software development process that mainly specifies what you should do to develop a software product.
- No specific software engineering practices are prescribed for developing the product; the team needs to decide how to do it.
- The requirements are captured as items in a “product backlog”; the product owner (client) sets the priorities for the items.
- The software product is developed in a series of iterations called “sprints”.
- Teams self-organize to determine the best way to deliver the product.

SCRUM: WORKFLOW



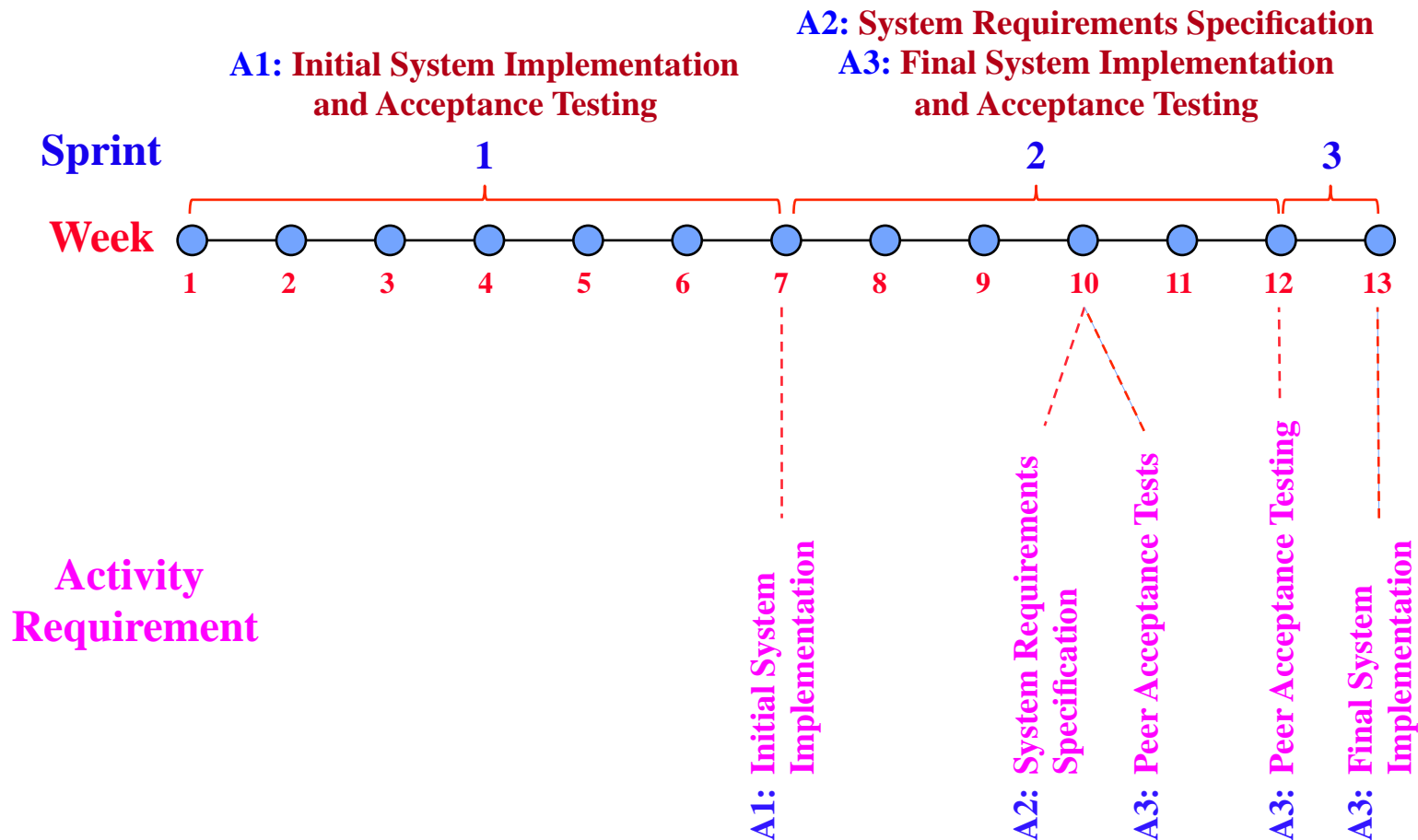
COPYRIGHT © 2005, MOUNTAIN GOAT SOFT

SCRUM: SPRINT

- A scrum project make progress in **a series of iterations** called “sprints” .
- The **typical duration** of a sprint is **2–4 weeks** or a calendar **month at most**.
- The software product is **designed, coded and tested** during the sprints.
- The requirements are **not allowed to change** during a sprint.

 **For the COMP 3111 project: There are three sprints.**

COURSE PROJECT SPRINTS



SCRUM: FRAMEWORK

Roles

- Product owner
- ScrumMaster
- Team

Meetings

- Sprint planning
- Daily scrum meeting
- Sprint review
- Sprint retrospective

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

SCRUM: PRODUCT OWNER (AKA CLIENT)

- Is the key stakeholder (represents users, client)
- **Defines** the requirements of the product.
- **Prioritizes** the requirements.
- **Adjusts** requirements and priority every iteration, as needed.
- **Decides** on the release date and content.
- **Accepts or rejects work results.**



SCRUM:

SCRUM MASTER (aka PROJECT MANAGER)

- Is the project team leader/person in charge (PIC).
- Is responsible for enacting **Scrum values and practices**.
- Ensures that the **team is fully functional and productive**.
- Enables **close cooperation across all roles** and functions.
- **Removes impediments** to progress.
- **Shields the team** from external interferences.



COMP 3111: TEAM REQUIREMENT

- Form your project team now.
 - Each team should have exactly 4 members.
- Arrange an initial project team meeting.
- Select a **name** and a **leader** (project manager) for your project team.

Submit

- Your project team name.
- The project team leader's name.
- Your project members' names and student numbers.

DEADLINE: Thursday, March 31 in the lecture.

SCRUM: FRAMEWORK

Roles

- Product owner
- ScrumMaster
- Team

Meetings

- Sprint planning
- **Daily scrum meeting**
- Sprint review
- Sprint retrospective

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

SCRUM: THE DAILY SCRUM

- A team meeting in which everyone answers three questions:

What did you do yesterday?

1

What will you do today?

2

Is anything in your way?

3

COMP 3111: MEETING REQUIREMENT

- Hold a weekly team (scrum) meeting in which everyone answers three questions:

1 What did you do in the **past week**?

2 What will you do in the **coming week**?

3 Is anything **hindering your progress**?

👉 **NOTE:** Only a weekly team (scrum) meeting is required.

COMP 3111: MEETING REQUIREMENT

- Your weekly team (scrum) meeting needs to have a:
 - meeting chair** who organizes and runs the meeting (usually the project manager).
 - minute recorder** who records the meeting minutes.
- Your team meeting minutes should record:
 1. The meeting location, date, time, present participants and absent participants.
 2. For each team member:
 - What he/she did since the last meeting (i.e., in the past week);
 - What he/she plans to do from now until the next meeting (i.e., in the coming week);
 - What impediments/problems hindered progress, if any;
 3. The date, time and place of the next meeting.

COMP 3111: MEETING REQUIREMENT

- Keep minutes of all your weekly team meetings.
- Distribute minutes to all team members soon after a meeting.

Submit

- All meeting minutes for the previous week.

**DEADLINE: Weekly on Thursday in the lecture.
(starting on Thursday, April 7).**

There is a required format for the meeting minutes available for download from the Project Resources course web page.

COMP 3111: FIRST TEAM MEETING

In your first team meeting you should:

- establish a common meeting time
- identify a meeting venue
- assign responsibility for taking minutes
- discuss areas of expertise
- construct a sprint backlog and burndown chart for Sprint 2

👉 **Hold your first team meeting before Thursday, April 7.**

SCRUM: FRAMEWORK

Roles

- Product owner
- ScrumMaster
- Team

Meetings

- Sprint planning
- Daily scrum meeting
- Sprint review
- Sprint retrospective

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

SCRUM: PRODUCT BACKLOG

- The **product backlog** represents the **requirements** of the system (i.e., **a list of all desired functionality** of the system).
- It is ideally expressed such that **each item has value** to the users or customers of the product.
- Items in the backlog are **prioritized by the product owner** (client).
- Items in the backlog are **reprioritized at the start of each sprint**.

SCRUM: SPRINT BACKLOG

- Items are selected from the product backlog and moved to the sprint backlog.
- Selection is usually based on item priority and on how much the team thinks they can do in a sprint.
- A product backlog item may be expanded into several sprint backlog tasks.
- Team members select the sprint backlog items to work on during the sprint.

👉 For the COMP 3111 course project:

The product backlog items for Sprint 2 are all the course project requirements not implemented in Sprint 1.

SCRUM: ESTIMATING

- Estimate the time for sprint backlog items **in hours using four discrete values** (1 day = 8 hours)
 - 1, 2, 4, 8 hours
- Round up in between estimates to the next highest discrete value.

 **For the COMP 3111 course project:**

Use Sprint 1 experience to make estimates for Sprint 2 tasks.

Also use **multiples of 8 hours** (i.e., 8 x number of days) for estimating.

SCRUM: BURNDOWN CHART

- Represents the amount of work (in hours) remaining in a sprint.
- For each task in the sprint backlog, the time required to complete it is estimated and summed.
- The chart is updated daily by estimating the work (hours) remaining.
- Allows the team to track progress and identify problems early.

SCRUM: EXAMPLE BURNDOWN CHART

Tasks	Mon	Tue	Wed	Thu	Fri
Code the user interface	8	4	4		
Code the middle tier	8	4	8	4	
Test the middle tier	8	4	2	4	2
Write online help	4		2		

The hours remaining for each sprint task can be kept in a spreadsheet where each cell estimates the remaining hours to complete the task.



The burndown chart graphically shows the total hours remaining each day to complete the sprint.

HOW TO CONSTRUCT A BURNDOWN CHART

Step 1: Determine tasks in the sprint.

Let's assume a 4 week sprint.


Sprint Tasks	Week 1	Week 2	Week 3	Week 4
Design database				
Design web pages				
Code web pages				
Test system				

HOW TO CONSTRUCT A BURNDOWN CHART

Step 2: Determine initial estimate of hours required to complete each task and sum.

Let's assume we estimate in hours.

Sprint Tasks	Week 1	Week 2	Week 3	Week 4
Design database	2			
Design web pages	4			
Code web pages	8			
Test system	12			
Total hours	26			

 **Try to be accurate, but don't worry about it.
You will get better the more you do it!**

HOW TO CONSTRUCT A BURNDOWN CHART

Step 3: Construct burndown chart.

The sum of hours for all tasks is the initial point on the chart and represents the total time that is estimated to be required to complete the entire sprint *at the start of Week 1*.



HOW TO CONSTRUCT A BURNDOWN CHART

Step 4: Revise task estimates at the beginning of each week.

Revised sprint backlog at the beginning of Week 2.

Sprint Tasks	Week 1	Week 2	Week 3	Week 4
Design database	2			
Design web pages	4	1		
Code web pages	8	4		
Test system	12	8		
Total hours	26	13		

Estimate of how many hours are still required to complete each task at the beginning of week 2.

HOW TO CONSTRUCT A BURNDOWN CHART

Step 4: Revise burndown chart at the *beginning of each week.*

Revised burndown chart at the beginning of Week 2.



HOW TO CONSTRUCT A BURNDOWN CHART

Step 4: Revise task estimates at the *beginning of each week*.

Revised sprint backlog at the beginning of Week 3.

Sprint Tasks	Week 1	Week 2	Week 3	Week 4
Design database	2		1	
Design web pages	4	1	2	
Code web pages	8	4	4	
Test system	12	8	8	
Total hours	26	13	15	

Estimate of how many hours are still required to complete each task at the beginning of week 3.

HOW TO CONSTRUCT A BURNDOWN CHART

Step 4: Revise burndown chart at the *beginning of each week.*

Revised burndown chart at the beginning of Week 3.



HOW TO CONSTRUCT A BURNDOWN CHART

Step 4: Revise task estimates at the beginning of each week.

Revised sprint backlog at the beginning of Week 4.

Sprint Tasks	Week 1	Week 2	Week 3	Week 4
Design database	2		1	
Design web pages	4	1	2	
Code web pages	8	4	4	2
Test system	12	8	8	4
Total hours	26	13	15	6

Estimate of how many hours are still required to complete each task at the beginning of week 4.

HOW TO CONSTRUCT A BURNDOWN CHART

Step 4: Revise burndown chart at the *beginning of each week.*

Revised burndown chart at the beginning of Week 4.



WHAT TO HAND IN EACH WEEK

Sprint Backlog and Burndown Chart

Sprint Tasks	Week 1	Week 2	Week 3	Week 4
Design database	2		1	
Design web pages	4	1	2	
Code web pages	8	4	4	2
Test system	12	8	8	4
Total hours	26	13	15	6



👉 For the course project you only need to construct and hand in a *weekly* burndown chart.

COMP 3111:

SPRINT BACKLOG & BURNDOWN CHART

- *For the course project,*
 - From the product backlog, create a **sprint backlog** for Sprint 2.
 - Create and maintain a **weekly burndown chart** for Sprint 2.
- Revise the sprint backlog *and* the burndown chart weekly.

Submit

- **initial sprint backlog and burndown chart on Thursday, March 31.**
- *revised sprint backlog and burndown chart weekly.*

DEADLINE: Thursday in the lecture.

There is a required format for the sprint backlog and burndown chart available for download from the Project Resources web page.