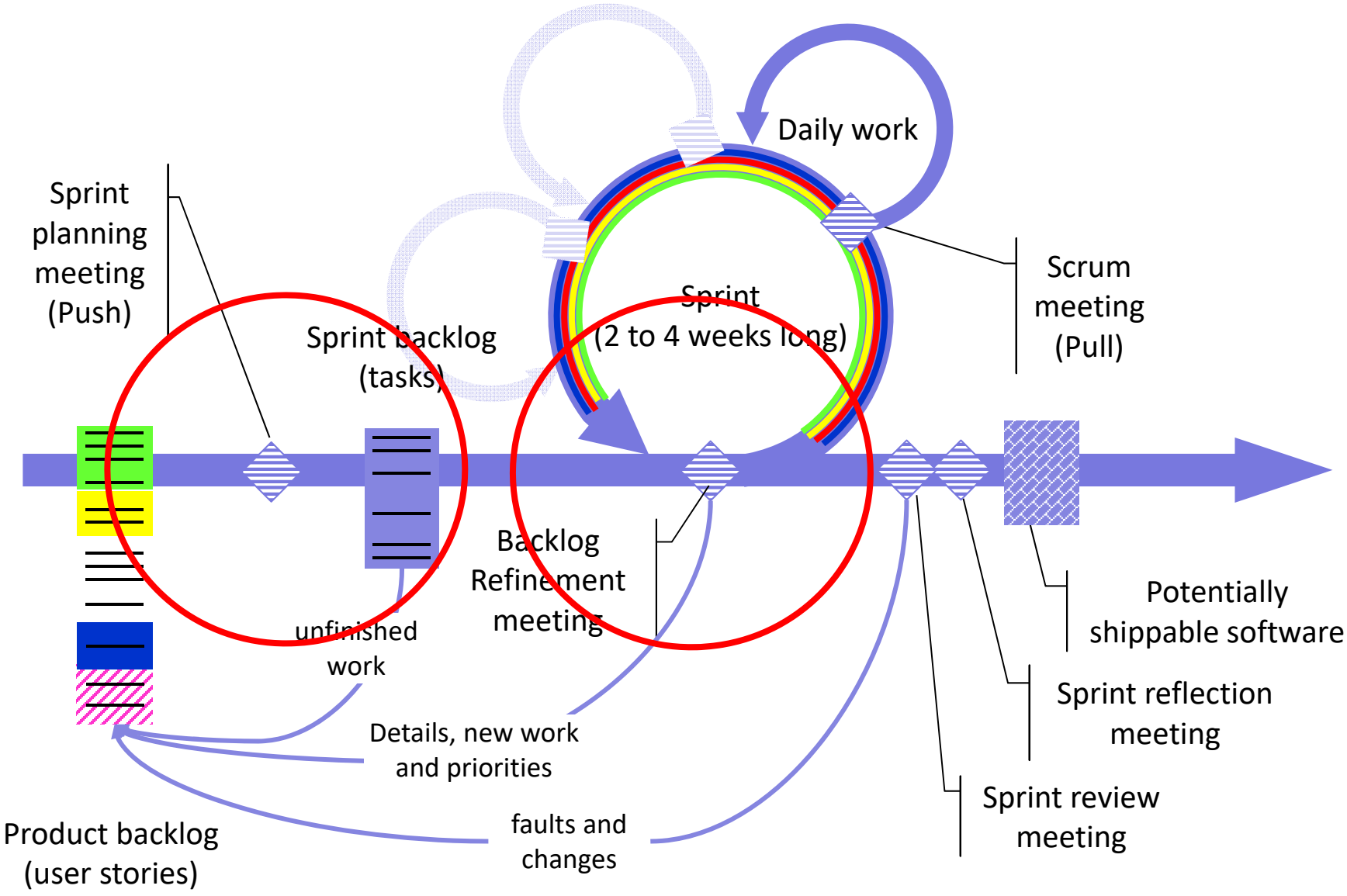
Sprint Planning & Backlog Refinement

Dr. Eduardo Miranda

Scrum


Daily work
Sprint planning meeting (Push)
Sprint backlog (tasks)
Sprint (2 to 4 weeks long)
Scrum meeting (Pull)
Backlog Refinement meeting
unfinished work
Potentially shippable software
Sprint reflection meeting
Details, new work and priorities
Sprint review meeting
Product backlog (user stories)
faults and changes

Sprint planning meeting (Traditional)

- Goal:
  - Plan the work for the coming iteration
- Activities:
  - Product owner selects stories to develop taking in account what was accomplished in the last Sprint and the business priorities (Sprint Planning Meeting Part 1)
  - Team breaks the selected stories down into the tasks necessary to realize them, estimates their effort and confirms viability based on the number of work hours available (Sprint Planning Meeting Part 2)
- When:
  - At the beginning of each iteration
- How long:
  - 1 to 2 hours per every week of planning
- Participants:
  - Product owner
  - Scrum master
  - Team members

Sprint backlog

Stories (From the PB)	Tasks	Estimates (hrs.)
As a member, I can read profiles of other members so that I can find someone to date	Prototype …	12
	Design UI …	4
	Code the …	12
	Test …	8
	Code …	10
	Test …	4
Update ABC to version 3.0	Configure …	16
	Migrate data …	8
	Conduct performance test …	8
As a member I can update my billing information	Design UI	8
	Review business rules	4
	Code …	16
Development effort (hrs.)		**110**

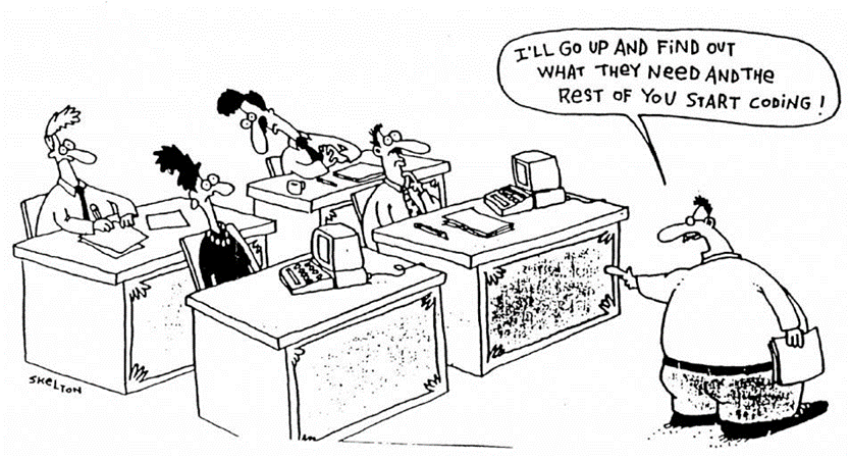
Select top priority PBIs

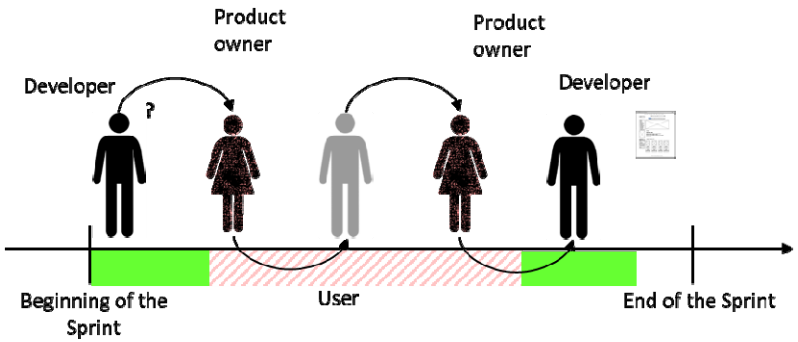
- What is wanted in the increment resulting from the upcoming Sprint?
  - Sprint goal
    - Examples: Support initial report generation; Load and curate North America map data; Demonstrate the ability to send a text message through an integrated software, firmware, and hardware stack; Implement basic shopping cart functionality including add, remove, and update quantities
  - Top priority user stories
    - Examples: As a job seeker I want to Login so I can use the system capabilities reserved to registered job seekers; As a job seeker I want to Logout to end a session and protect my data from being accessed by unauthorized people; As a job seeker I want to Register so I can make my data available to headhunters and use the system capabilities reserved for registered job seekers
  - Technical stories
    - Examples: We need to redesign the database to support the new requirement; we need to move to the new version of …
- What can be delivered in the increment resulting from the upcoming Sprint?
  - Velocity
    - Number of story points and capacity should be commensurate with past observations
  - Ready
    - All items on the DoR present or known

Items to be worked at an iteration need to be ready by the time the iteration is about to start

"Here's what happens if stories aren't ready. The Team is estimating and forecasting that they can finish vague and incomplete stories. They waste time and energy trying to get clarity from the Product Owner on exactly what the story means. People get frustrated and annoyed and run around in circles rather than getting down to work. Or that one vague story actually turns out to be five real stories once the work is actually begun. Or they work on the wrong thing, or the right thing in the wrong way, forcing the work to be re-done"

The Dangers of Not Being Done, Or Ready For That Matter; Jeff Sutherland, Feb 24, 2012





When PBIs are not ready, valuable time is wasted waiting for the answers

Definition of ready example

- User Story defined
- User Story Acceptance Criteria defined
- User Story dependencies identified
- User Story sized by Delivery Team
- User Experience artifacts are Done, and reviewed by the Team
- Architecture criteria (performance, security, etc.) identified, where appropriate
- Person who will accept the User Story is identified
- Team has reviewed the User Story
- Team knows what it will mean to demo the User Story

Definition of Ready: An Experience Report from Teams at Cisco, Ken Power, 2014

Map PBIs to implementation tasks

- For each selected PBI ask the team:
  - How do we build it?
  - How do we test it?
  - What do we need to update?
  - What do we need to document?
- Always verify that the defined tasks are sufficient to meet the conditions set forth in the definition of done

Task estimation

- Right size
  - All tasks are allocated the same number of hours, the "right size"
  - Tasks are grouped/split to add to the "right size"
  - In the burndown count number of tasks left
- Ideal hours
  - Estimate effort in ideal hours
  - Do not spend more time estimating than doing
  - Helps:
    - Development of a common understanding
    - Preventing over commitments
    - Reduce social loafing

Accept PBIs

Iteration backlog

Stories (From the PB)	Tasks	Estimates (hrs.)
As a member, I can read profiles of other members so that I can find someone to date	Prototype …	12
	Design UI …	4
	Code the …	12
	Test …	8
	Code …	10
	Test …	4
Update ABC to version 3.0	Configure …	16
	Migrate data …	8
	Conduct performance test …	8
As a member I can update my billing information	Design UI	8
	Review business rules	4
	Code …	16
Development effort (hrs.)		110

$$\sum_{\forall TasksInIteration} HoursPerTask \leq TeamCapacity$$

Team member	Days available - PTO	Nominal hours	Off-project hours	Other project activities	Member capacity
Mary	10	80	2	16	62
Jorge	10	80	5	16	59
Siva	8	64	0	13	51
Lee	10	40	0	8	32
				Team capacity	204 hrs

- Identify "anyone" and "expert" tasks
  - Anyone tasks are tasks that can be performed by whoever becomes available first
  - Expert tasks are tasks, that because the skills they required can only be performed by specific team members
- Compare the demands of the expert tasks against the availability of the experts
- Compare the demands of the anyone tasks against the availability of the team minus the expert commitments
- Verify that the internal dependencies do not impose a sequence of work whose duration exceeds that of the iteration

During iteration planning there is no explicit task scheduling nor allocation of “anyone” tasks

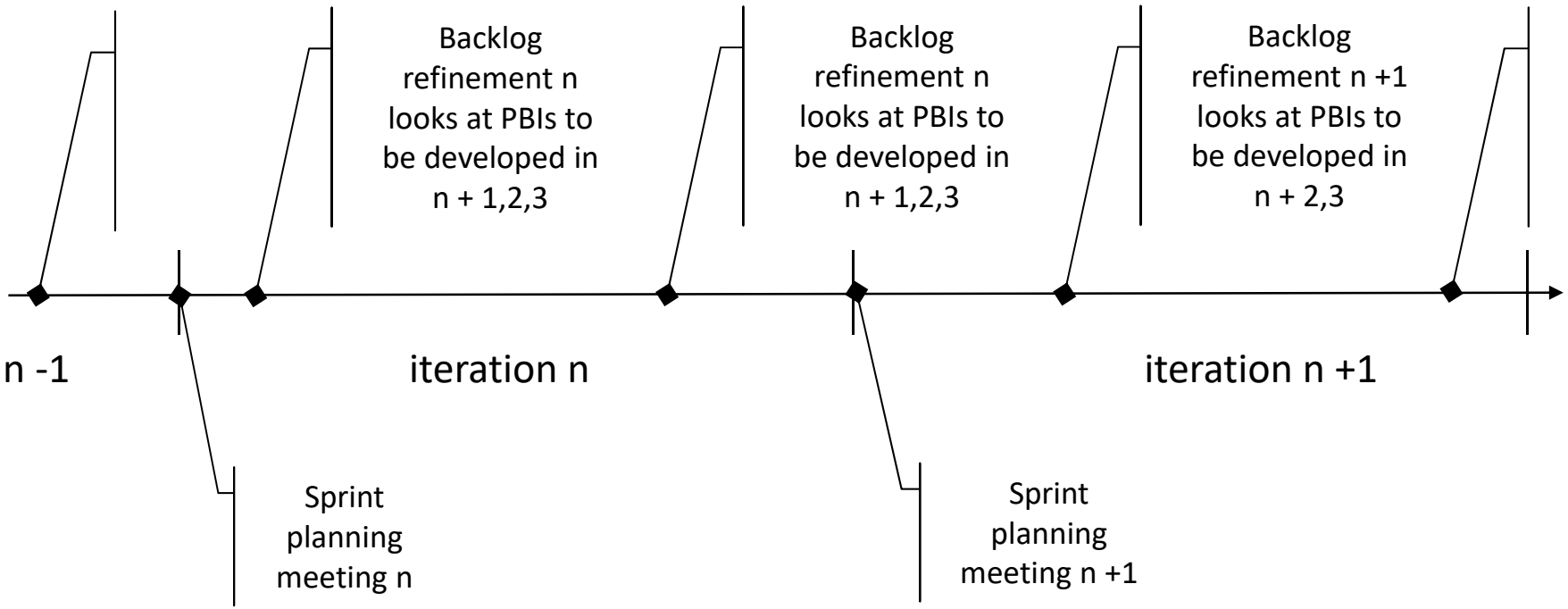
- No scheduling of anyone tasks: the team will decide what needs to do next during the daily iteration meetings, as this might depend on the state of other tasks
- No allocation of anyone tasks: everybody is collectively responsible, no concept of “I did my part”. Whoever becomes available will take the next most urgent task

Backlog refinement meeting

- Goal:
  - Look ahead planning. Give time to resolve things that will be done in future sprints (typically 1 or 2 sprints ahead of current meeting)
- Activities:
  - Review forthcoming user stories
  - Add details (the conversation part) to user stories
  - Split user stories that appear too big to fit in a sprint
  - Create new user stories
  - Remove those that are not longer relevant
  - Reassess the relative priority of stories
  - Assign/correct estimates
- When:
  - Twice per iteration. Once to raise issues, once to discuss resolution or further actions
- How long:
  - 2 hours each session
- Participants:
  - Product owner
  - Scrum master
  - Team members
  - Other stakeholders

Backlog refinement meeting schedule

New PBIs, readying actions for existing PBIs, reprioritization



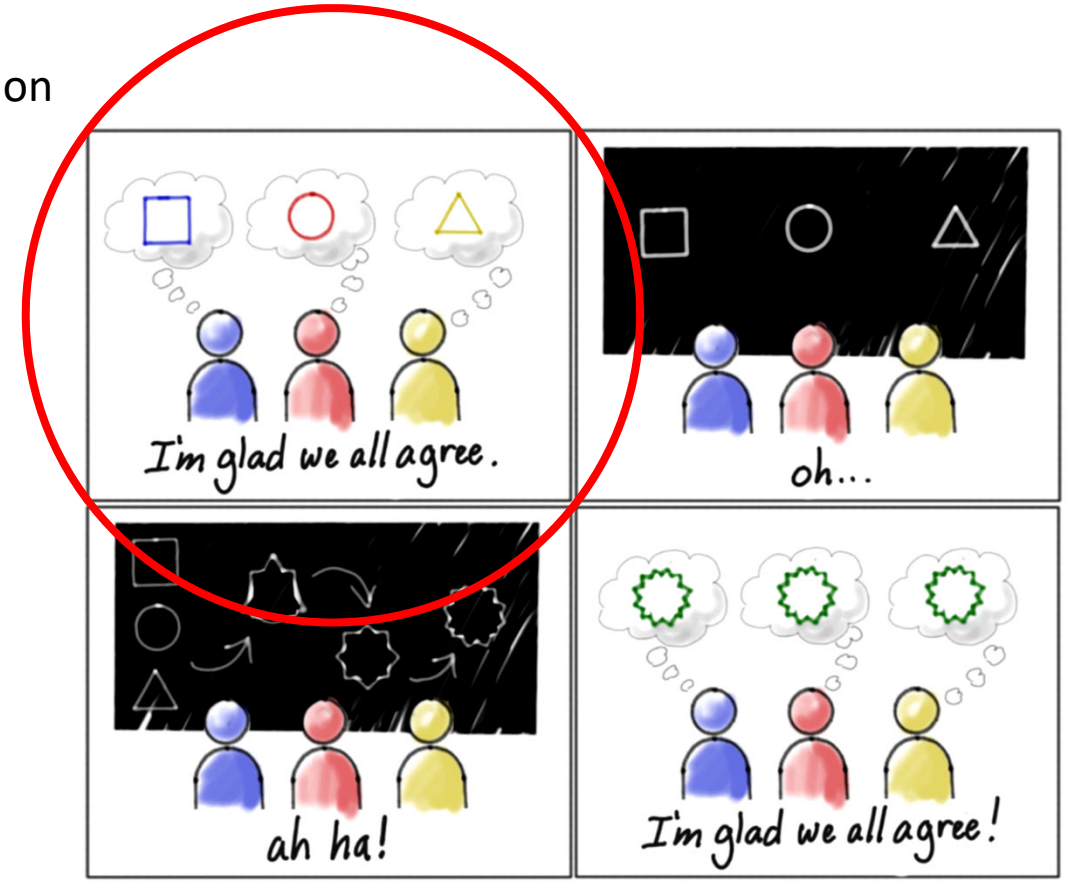
Only PBIs that satisfy the definition of ready

Reviewing PBIs

- Identify upcoming work
- Review PBIs against their respective definitions of ready
- Identify and assign action items
- If necessary create new PBIs
- Follow-up on previously created action items

The way Sprint planning is being practiced

Map PBIs to implementation tasks



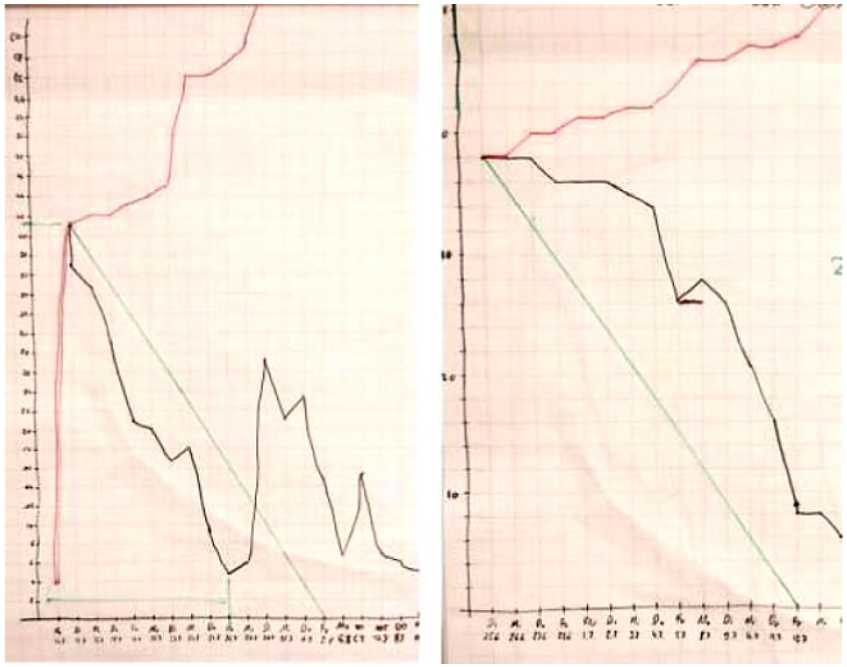
This is the root cause of lots of unplanned tasks and rework

30 to 40% - Schwaber and Sutherland, Scrum Guides, 2010

26% - Grapenthin, et al, Facilitating Task Breakdown in Sprint Planning Meeting 2, 2014

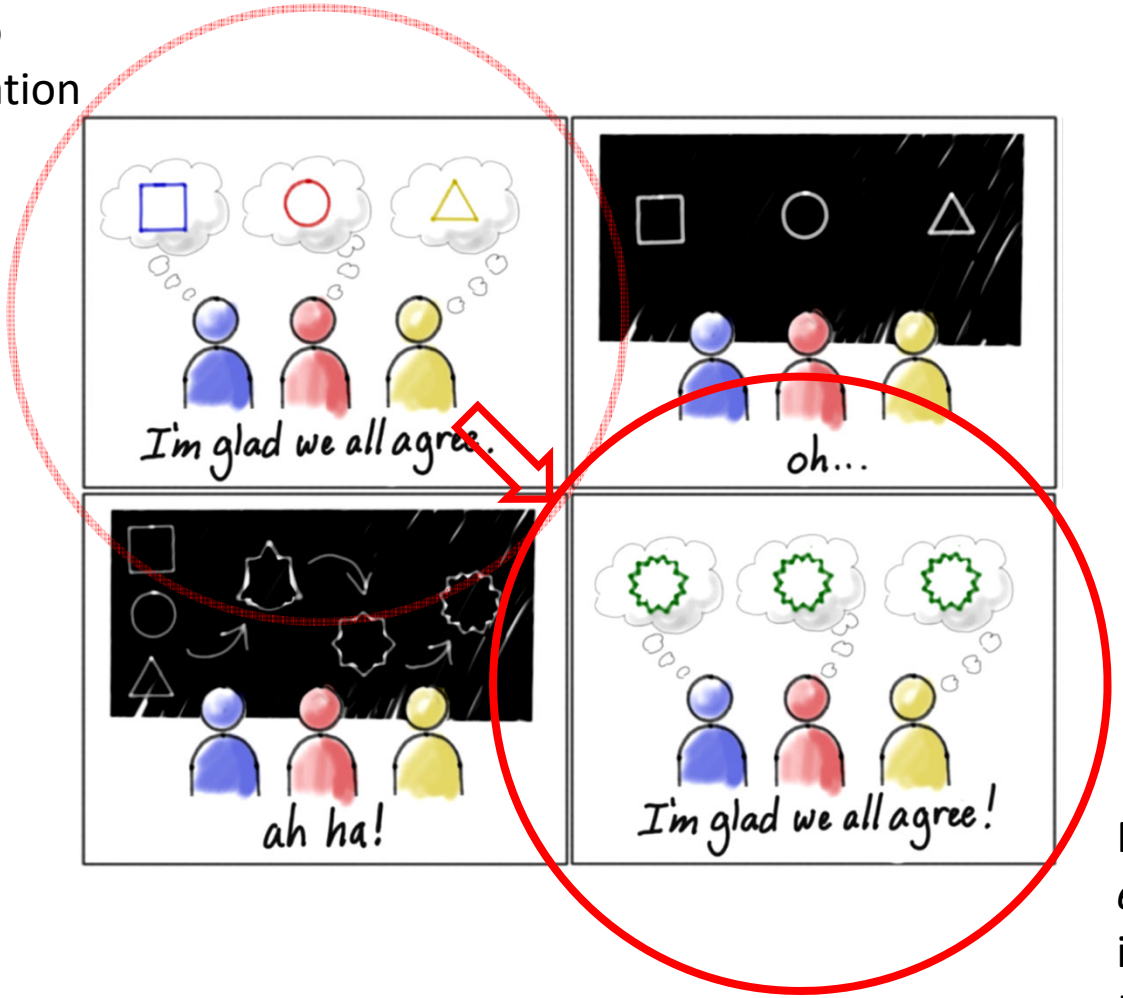
TIP

Usually, only 60-70% of the total Sprint Backlog will be devised in the Sprint Planning meeting. The rest is stubbed out for later detailing, or given large estimates that will be decomposed later in the Sprint.



Design Driven Sprint Planning

Map PBIs to implementation tasks



Map *design elements* to implementation tasks

Design Driven Sprint Planning: Full day

Existing design

Definition of ready

Sprint planning

Sketch the design

Select top priority PBI

Map to imp. tasks

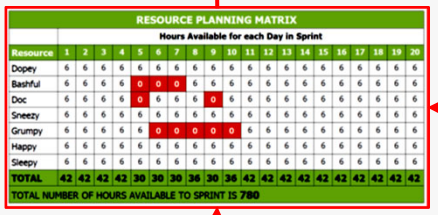
Estimate tasks

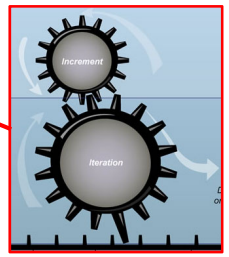
Accept PBI

Establish capacity

Product backlog items	Estimates
User $story_1$	Story points
User $story_2$	Story points
Technical $story_{n-1}$	Hours
User $story_n$	Story points

Sprint backlog items	Tasks	Estimates	Status
User $story_1$	T_{11}	Hours	
	T_{12}	Hours	
User $story_4$	T_{41}	Hours	
	T_{42}	Hours	
	T_{43}	Hours	
Technical $story_{n-1}$	$T_{n-1,1}$	Hours	
	$T_{n-1,2}$	Hours	
	$T_{n-1,3}$	Hours	
	$T_{n-1,4}$	Hours	





Velocity

Unfinished & team's PBIs

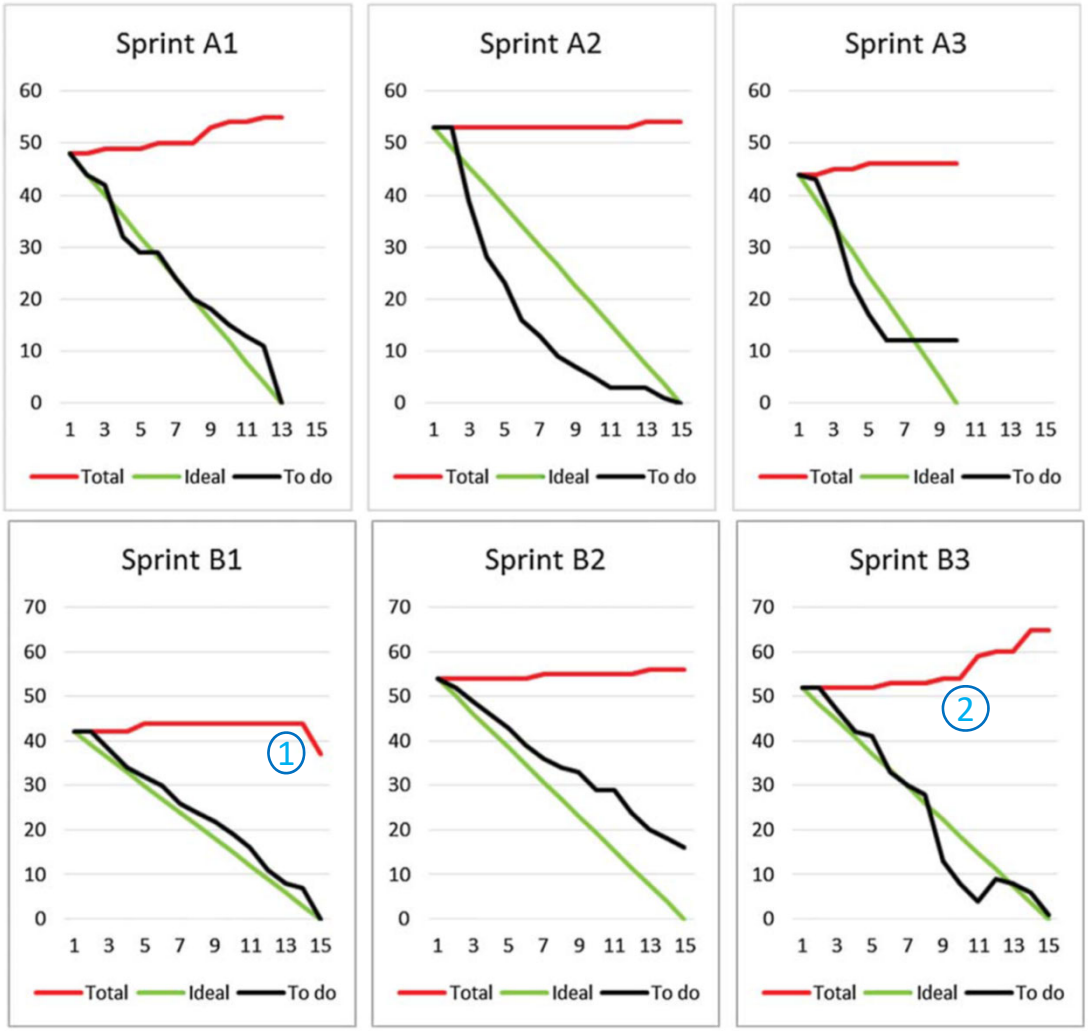
Sprint execution

Scrum meeting

Reduction in the number of unplanned tasks in a design based Sprint

Project A

Project B

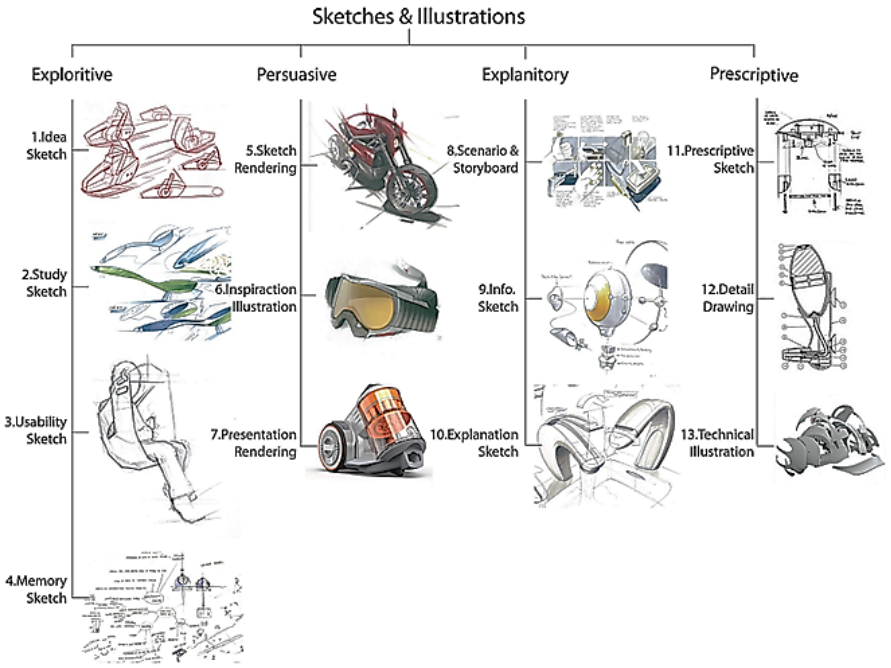


1. User story drop
2. Additional user stories

S. Grapentin et all, Improving task breakdown comprehensiveness in agile projects with an Interaction Room, 2015

Sketch the design

- A sketch is a rapidly executed freehand drawing that is not usually intended as a finished work. A sketch may serve a number of purposes: it might record something that the artist sees, it might record or develop an idea for later use or it might be used as a quick way of graphically demonstrating an image, idea or principle (Wikipedia)
- Model sketches should remain deliberately abstract, informal and incomplete, as they are not intended to replace formal system specifications. Rather, they should serve as a catalyst for the discussion and joint understanding of a project and its possible pitfalls (Graphentin)



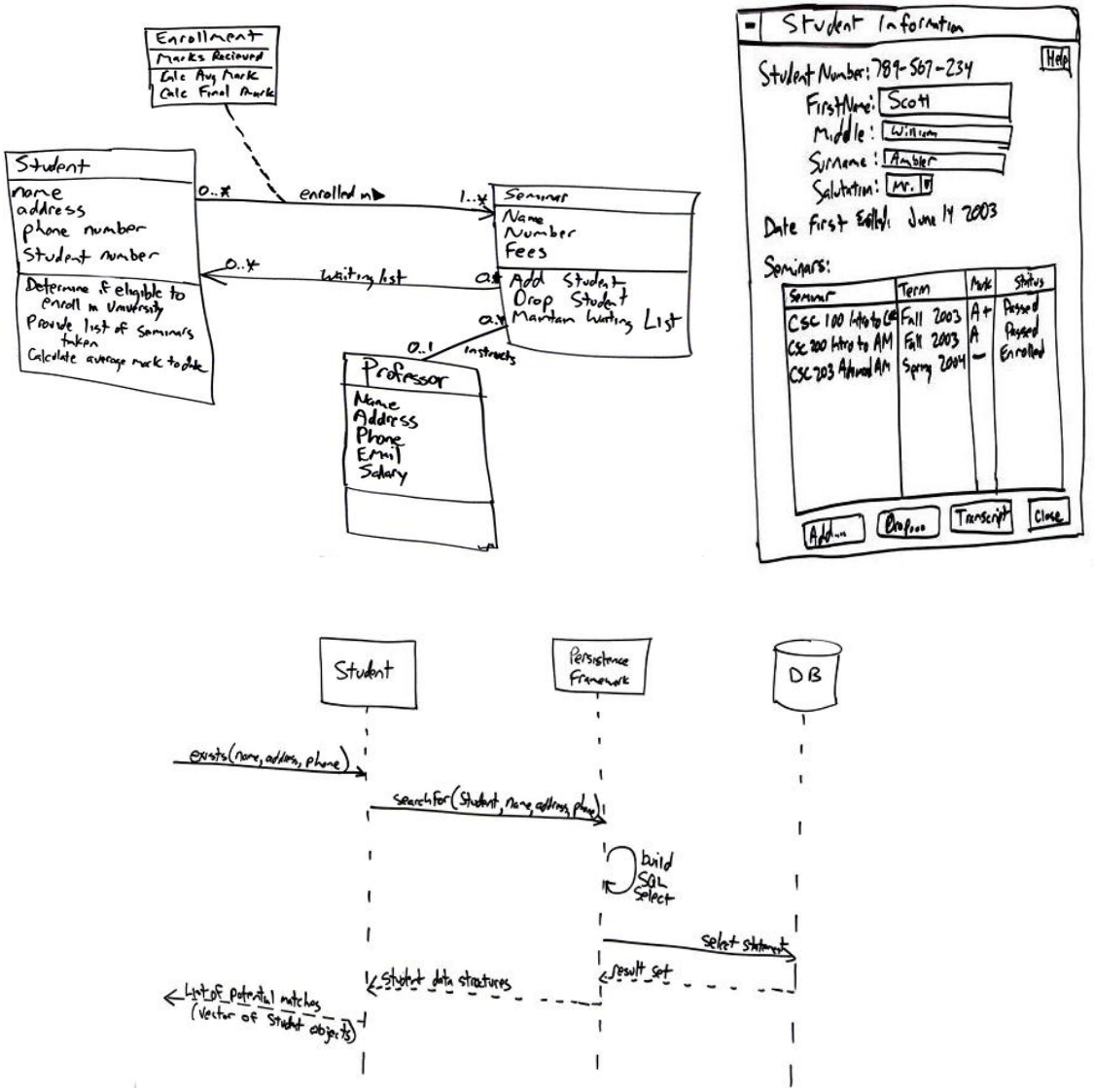
S. Grapentin et all, Improving task breakdown comprehensiveness in agile projects with an Interaction Room, 2015
J. Self, Why is Sketching (Still) Important (To Design)?, 2016

Sketching vs. specifications

- *Sketches fulfil their purpose (and no more).* For example, if the purpose is for John to explain to Steve and Dave how objects could interact to produce certain functionality, maybe all its needed is a sequence diagram
- *Sketches are understandable* to their intended audience but not necessarily to everyone. That is, the level of detail and the content needs to be appropriate for Steve and Dave, but not necessarily to any software engineer from any other project
- *Sketches are not a prescription for how to build the code*, but any coding decision that might contravene them should be brought to the consideration of others

Explicit design example: Agile modeling

"At the beginning of each iteration an agile team will typically plan (estimate and schedule) the work that they will do that iteration. To estimate each requirement accurately you must understand the work required to implement it, and this is where modeling comes in. You discuss how you're going to implement each requirement, modeling where appropriate to explore or communicate ideas. This modeling in effect is the analysis and design of the requirements being implemented that iteration"



Agile modeling, S. Ambler, accessed; 2019

Questions?