

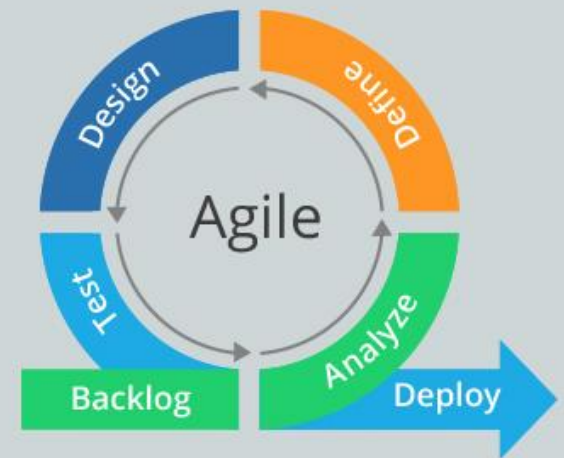
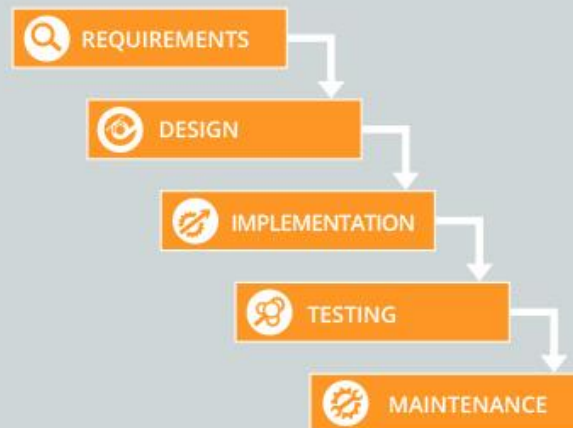
Agile! Scrum!

As implemented for CS@Mines Field Session

Design Methodologies

- Waterfall
- Agile
- And others

Waterfall vs. Agile



Agile in a Nutshell

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

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.

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

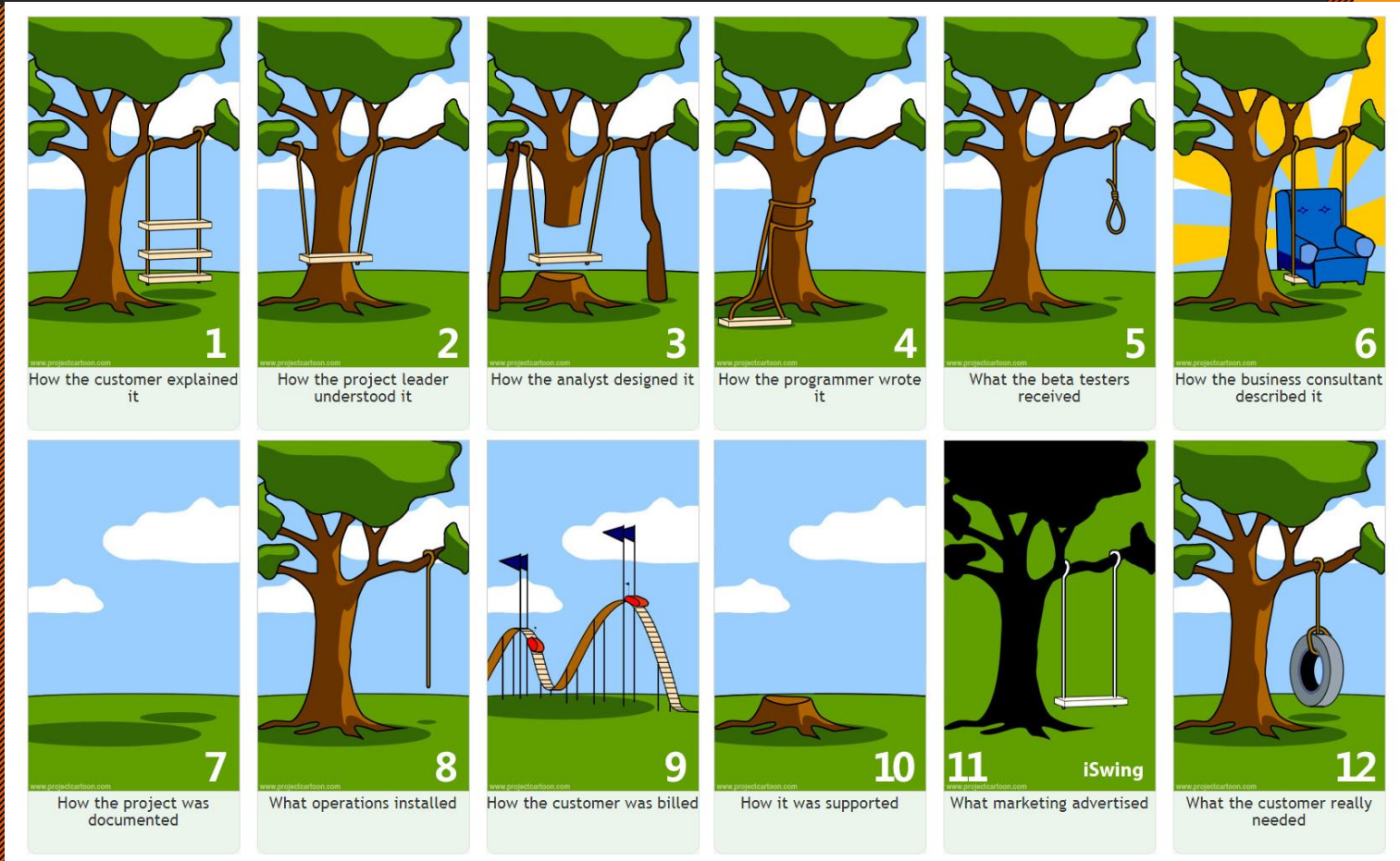
Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

© 2001, the above authors
this declaration may be freely copied in any form,
but only in its entirety through this notice.

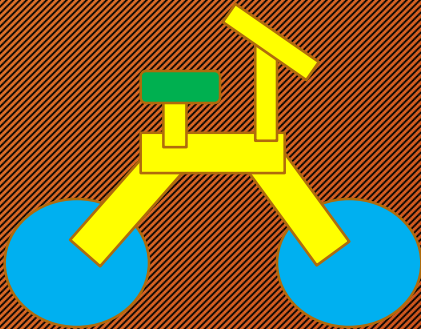
12 Agile Principles

<https://agilemanifesto.org/principles.html>

Waterfall



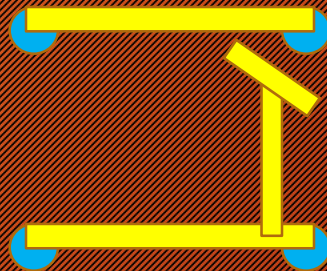
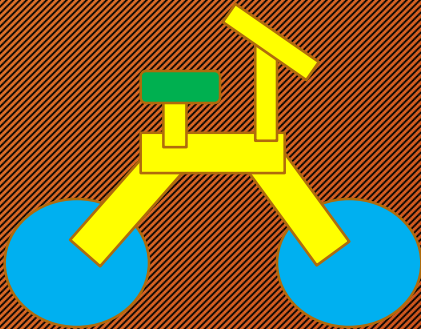
Agile



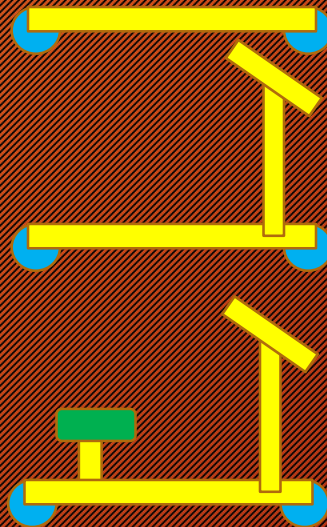
Agile



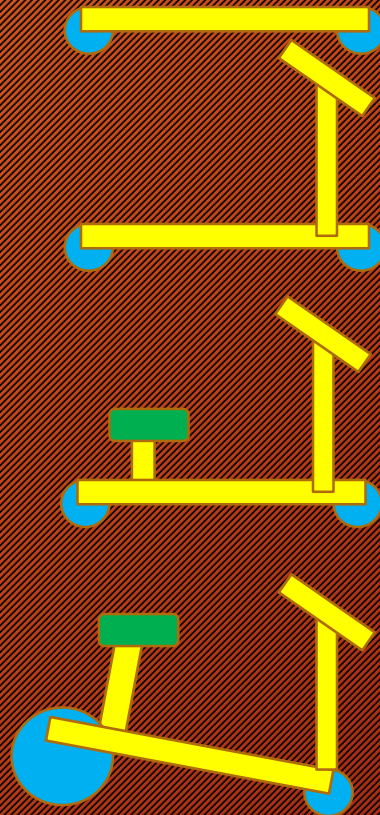
Agile



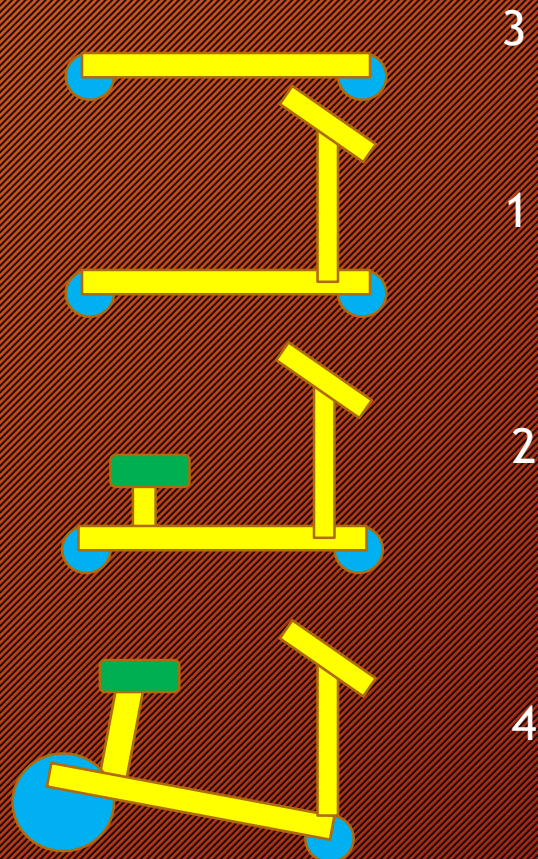
Agile



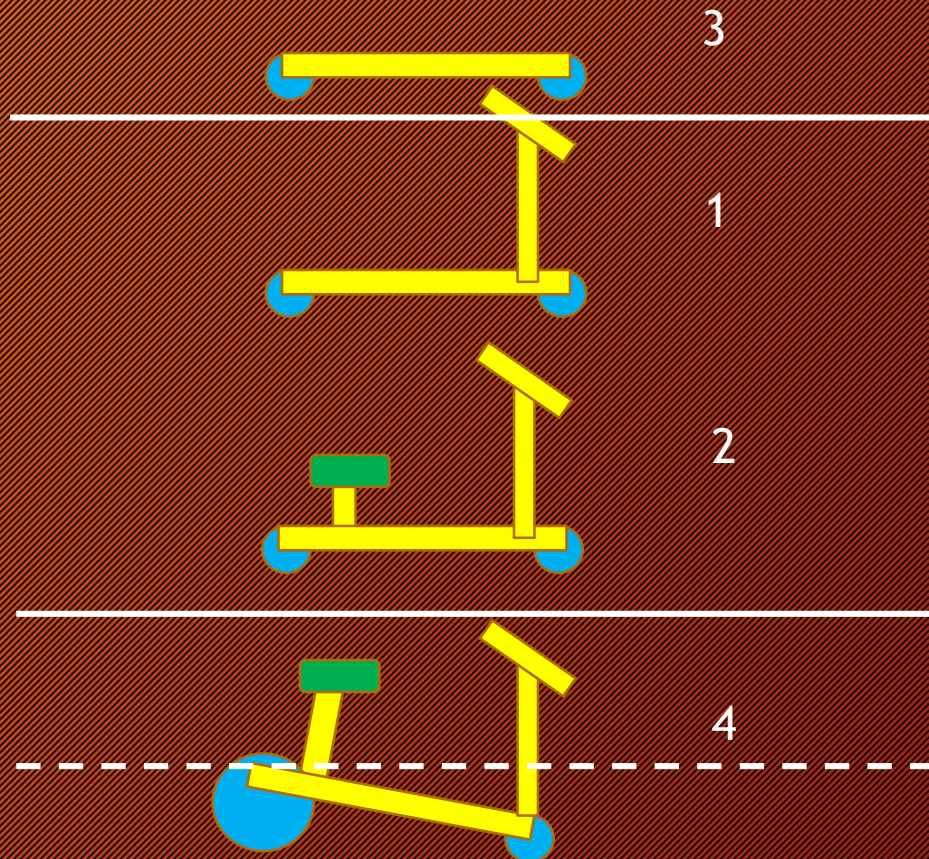
Agile



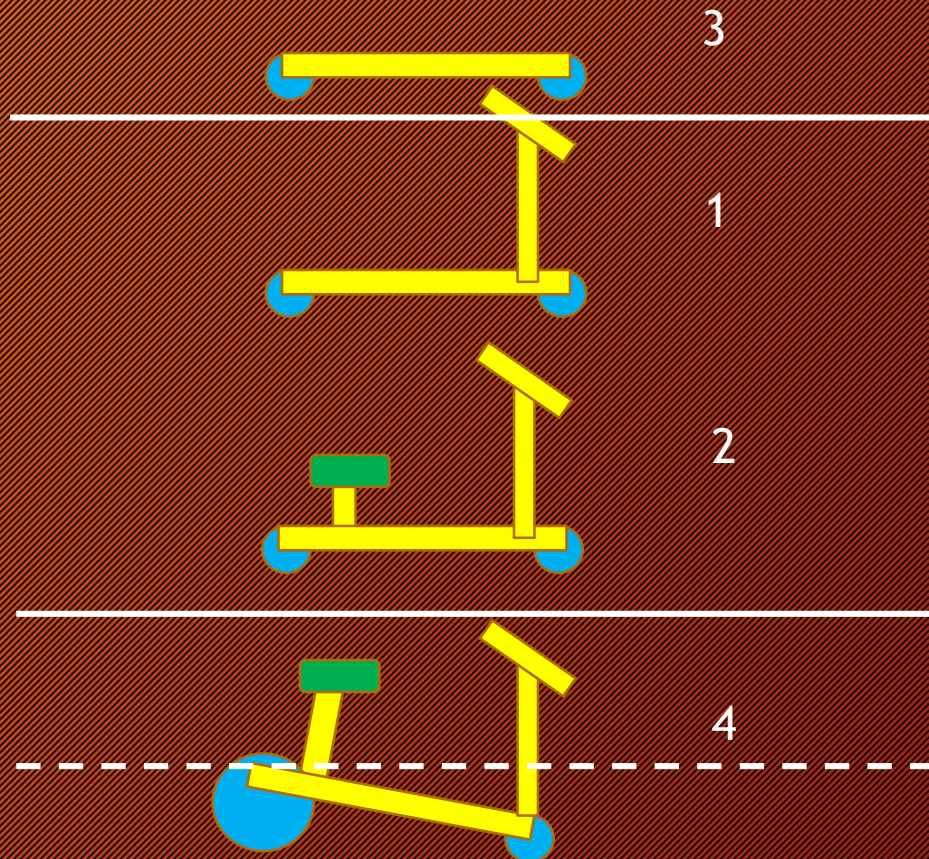
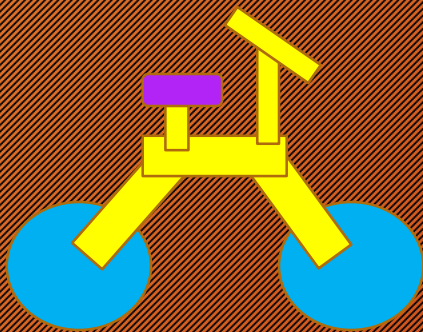
Agile



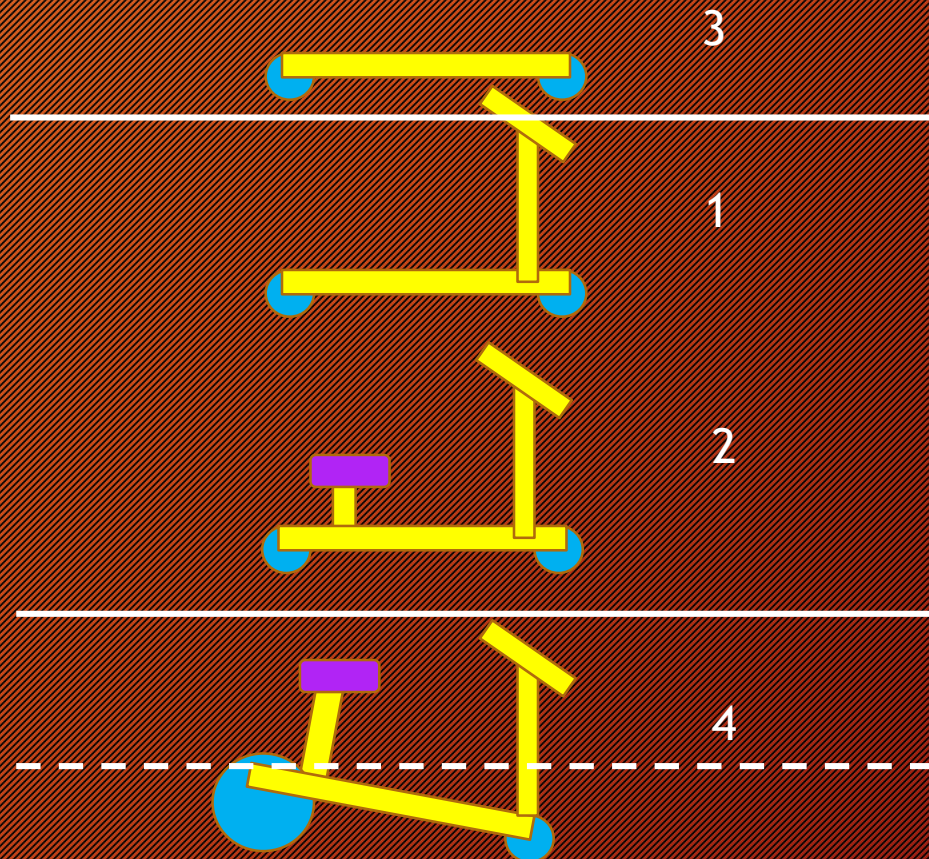
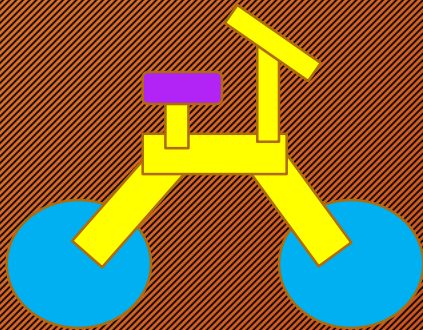
Agile



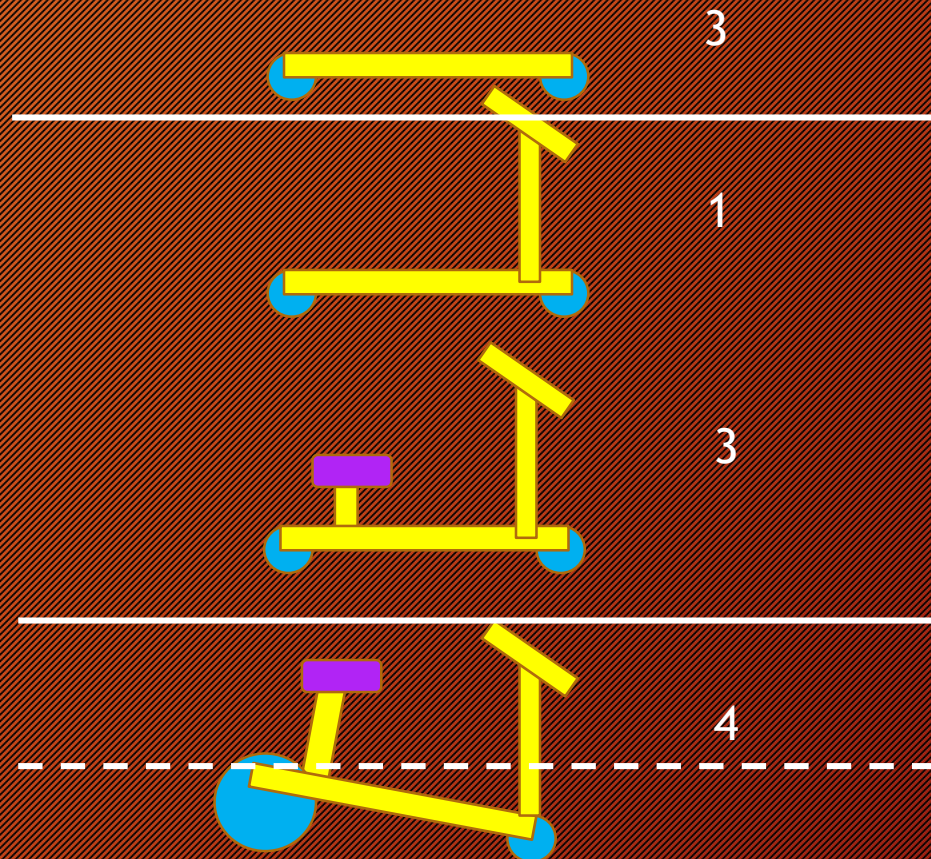
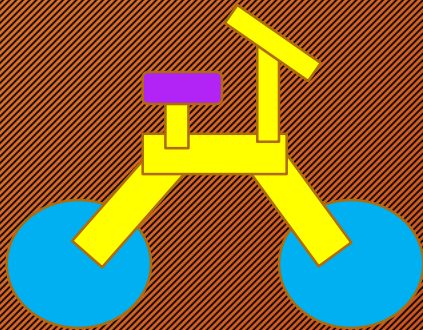
Agile



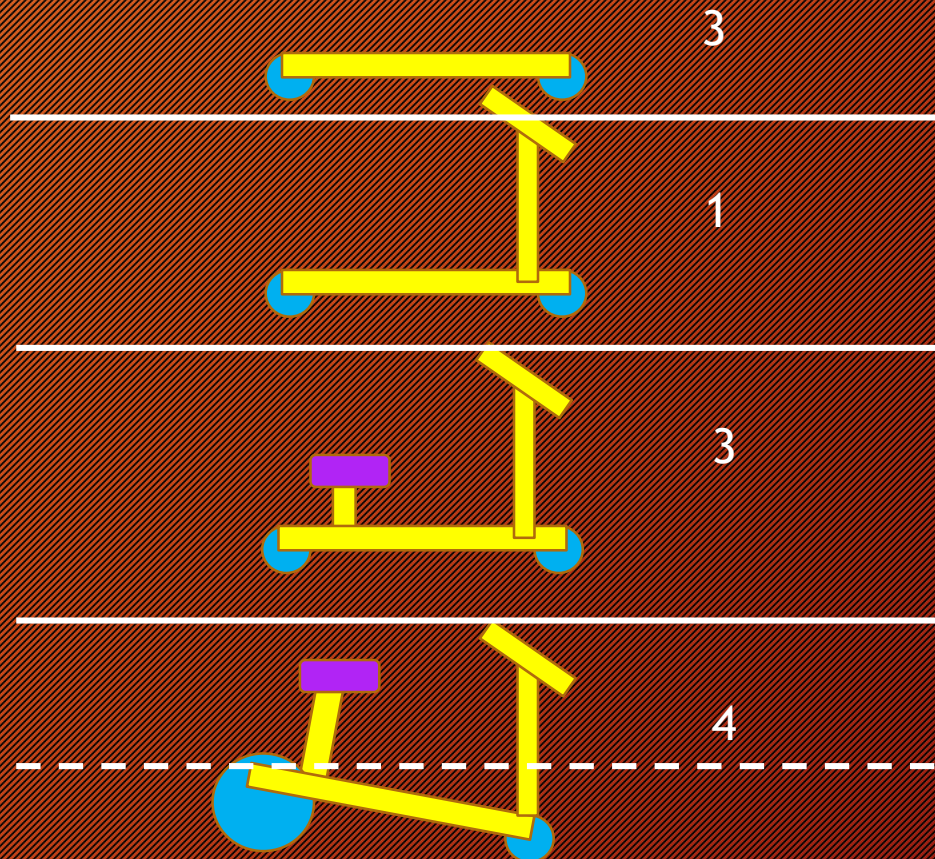
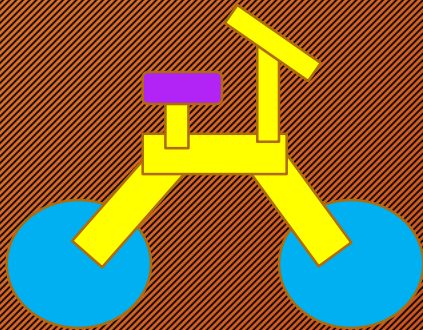
Agile



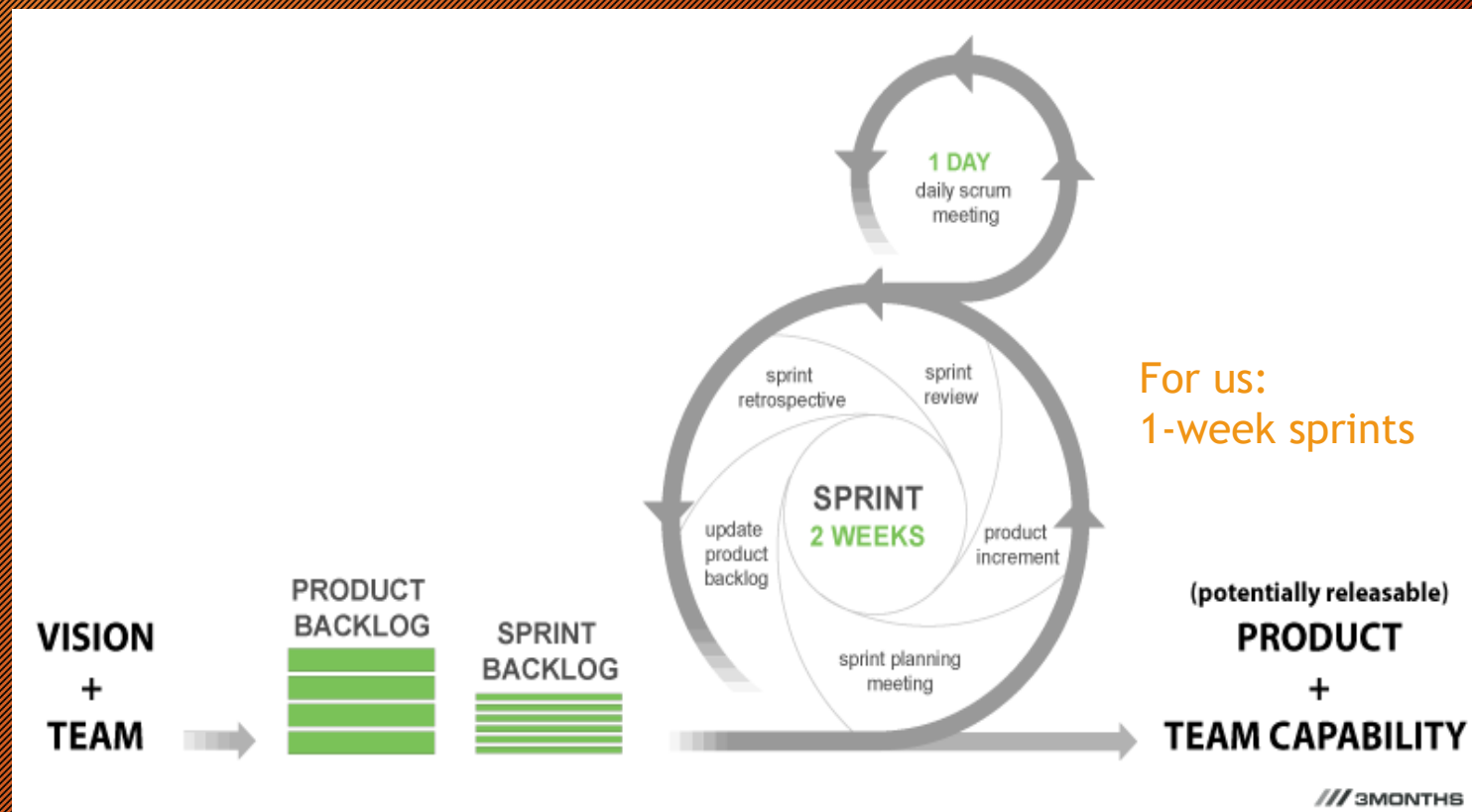
Agile



Agile



Scrum



Scrum meetings

the ongoing process

Team Meetings Each Sprint

- A. Sprint Planning Part One
- B. Sprint Planning Part Two
- C. Daily Scrum
- D. Sprint Review
- E. Sprint Retrospective

Advisors will attend some meetings, but for the most part it's up to the team to follow the Scrum process. Past teams have found it effective – try it!

Sprint Planning Part One

(start of sprint, team with client)

- Goal:
 - Agree on the work to be done in the next sprint
 - Focus on “what”
- Ask client to:
 - Approve use cases (unless client provided stories)
 - Prioritize
- When:
 - This should be done at start of each sprint. (Can be combined with Sprint Review at end of sprint.)

Agile principle:

“Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.”

Sprint Planning Part Two (team)

- Focus on “how”
- Team selects items to complete during sprint
 - Should be based on client priorities (unless changes are needed due to technical constraints)
- Team creates estimates of time to complete
- Team commits to get that work done
- ScrumMaster ensures scrum products are maintained
 - want to be able to review progress charts

Agile principle:

“Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.”

Daily Scrum (team)

- Short (~5 minute) meeting each work day
- Everyone on team attends
- Everyone remains standing
- Each member reports exactly 3 things:
 1. What they got done since last meeting
 2. What they are planning to finish by next meeting
 3. Any blocks or impediments

Brief description of blocks/impediments –
detailed discussion follows after scrum

Agile principles:

“The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.”

“The best architectures, requirements, and designs emerge from self-organizing teams.”

Daily Scrum (continued)

- Not a report to a manager
- Purpose is for team to self organize
- *No discussion during scrum.* If discussion needed, have follow-up meeting
- Generally recommended that managers *not* attend Daily Scrum
- **FOR US:** advisors will observe a daily scrum during sprint 2, but this is generally up to the team to do. Past teams have found this *very helpful*.
- Teams working onsite may just participate in client's daily scrum.

Daily Stand Up



Sprint Review

(end of sprint, team with client)

- Key idea: inspect and adapt the *product*
- In-depth conversation between team and client
- Includes a demo, along with discussion
- Not a presentation - no slides
- Guideline: no more than 30 minutes should be spent preparing for the review
- **For Us:** quick demos of product done with *client*, every sprint if possible. Occasional demos of product may be done for advisors (but generally it's the client who accepts/rejects).
- This will likely be combined with planning the next sprint (so one meeting with client per sprint)

Sprint Review (continued)

Agile principles:

“Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.”

“Working software is the primary measure of progress.”

“Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.”

Sprint Retrospective (end of sprint - team + advisor)

- Inspects and adapts the *process*
- What's working... and what's not
- Each team member adds items to both lists
- Adjust the process as needed
- **FOR US:** advisors will facilitate sprint retrospectives each sprint. If used properly, the retrospective should help to avoid team issues.
- We will also have team evaluations (more later)

Agile principle:

“At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.”

Scrum roles

who will do what, and how to keep it together

Scrum Roles

- Product Owner
- ScrumMaster
- Team

Scrum Roles - Product Owner

- Identify product features
- Prioritize the list
- Ensure product provides value
- Can be the customer; or the customer may be millions of people
- **FOR US:** client will identify/prioritize features
- We will also have one team member who is the primary contact for client, be sure to represent the client's needs during team discussions (i.e., *fully* understand goals/needs)

Scrum Roles - ScrumMaster

- Not the project manager (don't assign tasks)
- Guides the team in skillful use of Scrum
 - Must be process oriented!
- Maintain scrum reports
- Sometimes a dedicated role
- **FOR US:** part-time role, will swap roles if possible (depends on team size). Think about: who on your team always ensures the team gets the work done and meets the entire rubric. Pick that person. 😊

Scrum Roles - Team

- Cross-functional: includes all expertise necessary to deliver shippable product
- Self-organizing
- Typically 7 +/- 2 people
- All members should be 100% dedicated to work for one product during the Sprint
- Known as “pigs” (old joke)



Scrum Roles

Agile principles:

“Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.”

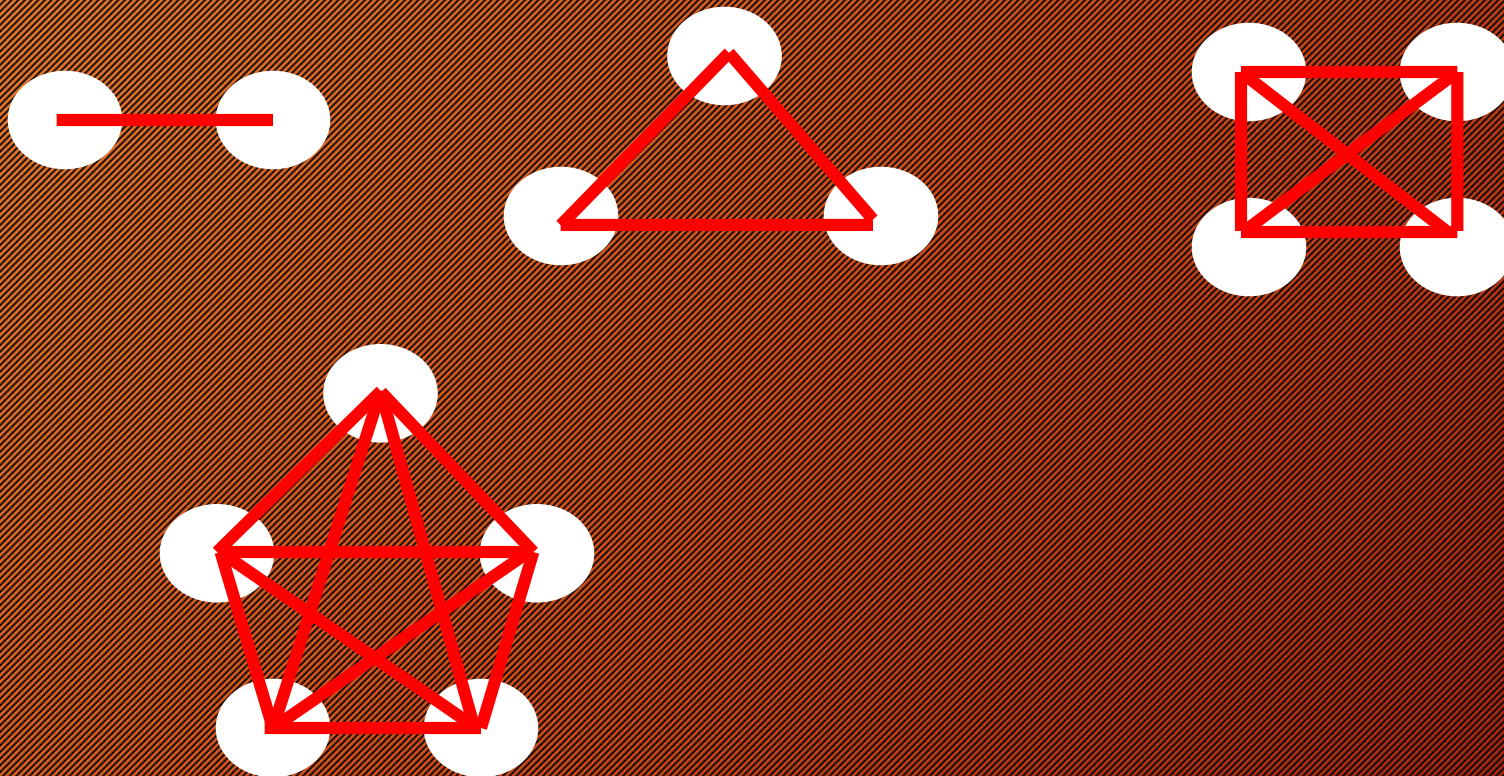
“Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.”

“Business people and developers must work together daily throughout the project.”

Team Makeup - Pair Programming

- Driver & Navigator
 - No distractions
 - No cowboyism
 - Constant code review
 - Sustainable
 - Don't get stuck

Rotate Pairs



Test-Driven Development

- Write a *failing* test
- Write code to pass the test
- Repeat

Run all tests frequently to ensure nothing has broken. Early fixes are cheaper.

In our field session timescale, it will be tempting to skip this important step. Don't. TDD done right will save you time (and you need the practice).

Agile principles:

“Continuous attention to technical excellence and good design enhances agility.”

“Simplicity--the art of maximizing the amount of work not done--is essential.”

Getting started with scrum

what to do this sprint

Make a team contract

More on this shortly...

Create Requirements document

- Full details online
- Process & some expansion to follow

Meet with the client (and follow-up)

- Show up on time
- Dress appropriately - hygiene is important
- Everyone should take notes!
 - People will focus on different things, this ensures you capture most of what the client says
- Look up vocabulary. Do some research. The best way to avoid being overwhelmed is to be specific about what you don't know and develop a plan to address.
- Create a list of questions (if needed). Have your client contact send questions to ask for clarification.
- Bottom line: do some investigation on your own, then don't assume
- ASK!

Starting Scrum

Based on interviews with client:

- Create Product Backlog
- Write Use Cases (or acquire stories)
- Initial estimate of effort

This sprint, you'll create a requirements document before you meet with your advisor. By the end of the sprint you should have the initial use cases or stories.

Effort Estimation - Option 1

Points as a *measure of effort*:

- Effort refers to amount of work
- Units are time-based, e.g., person-days
- Usual definition is one point = 8 hours (perfect person-day)
- Velocity is based on number of hours available. Rule of thumb is about 75%, so ~6 hours of an 8 hour day.

Interesting!

From <http://deepscurm.wordpress.com/2009/11/12/a-brief-defense-of-time-estimating-sizes-for-scrum-projects/>

Effort Estimation - Option 2

Points as a *measure of complexity*:

- No standard means to define complexity
- Use a relative scale, sometimes compared to t-shirt sizes etc. (Fibonacci numbers are a popular scale)
- Velocity is based on team history

More thoughts:

<http://www.velocitycounts.com/2013/05/why-do-high-performing-scrum-teams-tend-to-use-story-point-estimation/>

Effort Estimation - Field Session

- Key criteria: *Reliability of predictions*
 - i.e., can the team predict how long it will take to complete their user stories
- Per blog author: “I have seen no compelling evidence that either approach is superior for the key measure”
- Effort estimation based on time is typically easier for team members to grasp in short timeframe
- Suggest you use time-based effort estimation
 - But use half-days rather than days as your unit

Scrum artifacts

Scrum Artifacts

- Product Vision
- Product Backlog
- Release Plan
- Sprint Backlog
- Sprint Burndown
- Impediment List (not required)

Maintained by Scrum Master, advisors will review briefly, but it's up to the *team* to ensure the product is successfully delivered by the end of field session.

Product Vision

“Often Scrum’s emphasis on ‘getting work done’ is misunderstood as a rush to develop with not enough thought to where the project should be going. Don’t make that mistake. Every Scrum project needs a product vision that acts as the project’s true north, sets the direction and guides the Scrum team.”

Vision for us

- Ensure that you have a good understanding of client's goals during sprint 1
- Product vision is included in your requirements document
- Brief client and product description will also be included in the final report

Product Backlog

	Item #	Description	Est	By
Very High				
	1	Finish database versioning	16	KH
	2	Get rid of unneeded shared Java in database	8	KH
		- Add licensing	-	-
	3	Concurrent user licensing	16	TG
	4	Demo / Eval licensing	16	TG
		Analysis Manager		
	5	File formats we support are out of date	160	TG
	6	Round-trip Analyses	250	MC
High				
		- Enforce unique names	-	-
	7	In main application	24	KH
	8	In import	24	AM
		- Admin Program	-	-
	9	Delete users	4	JM
		- Analysis Manager	-	-
		When items are removed from an analysis, they should show up again in the pick list in lower 1/2 of the analysis tab	8	TG
		- Query	-	-
	11	Support for wildcards when searching	16	T&A
	12	Sorting of number attributes to handle negative numbers	16	T&A
	13	Horizontal scrolling	12	T&A
		- Population Genetics	-	-
	14	Frequency Manager	400	T&M
	15	Query Tool	400	T&M
	16	Additional Editors (which ones)	240	T&M
	17	Study Variable Manager	240	T&M
	18	Haplotypes	320	T&M
	19	Add icons for v1.1 or 2.0	-	-
		- Pedigree Manager	-	-
	20	Validate Derived kindred	4	KH
Medium				
		- Explorer	-	-
		Launch tab synchronization (only show queries/analyses for logged in users)	8	T&A
	22	Delete settings (?)	4	T&A

Should include:

- Item description
- Priority
- Point estimate
- Name of use case or story

Create with:

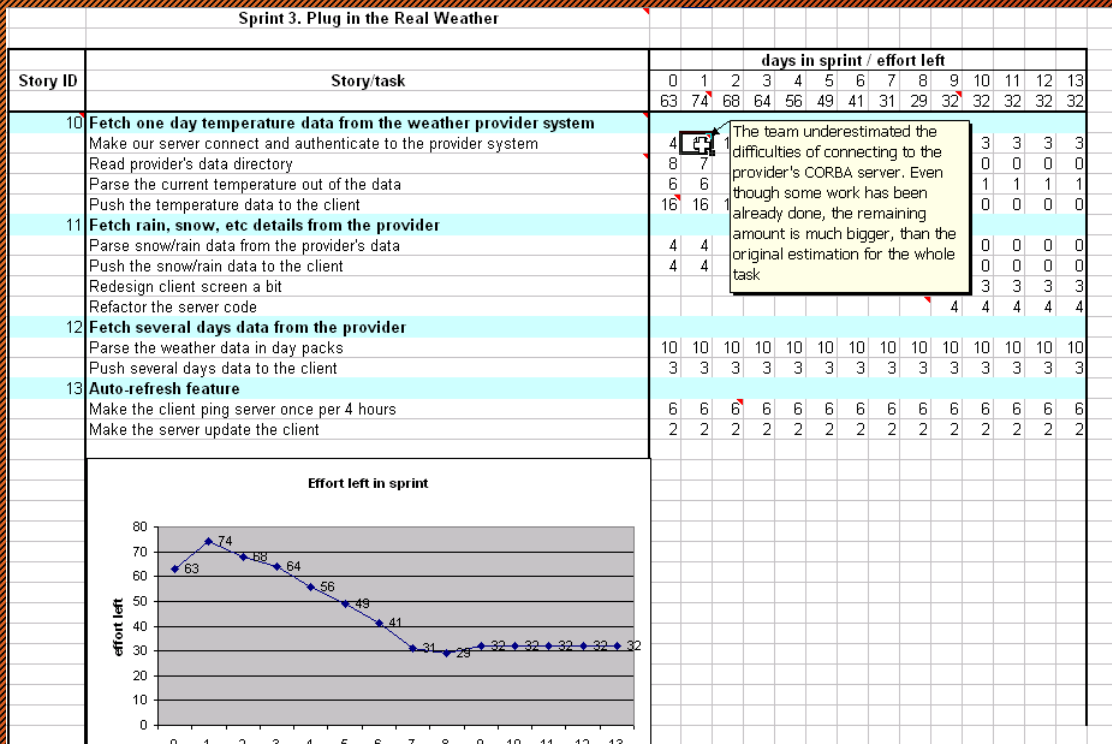
- Spreadsheet
- Free tool

Many varieties, this one from:

<http://www.agile-tools.net/backlog>

Release Plan/Sprint Backlog

- Subset of the Product Backlog in current release
- Includes additional detail about tasks

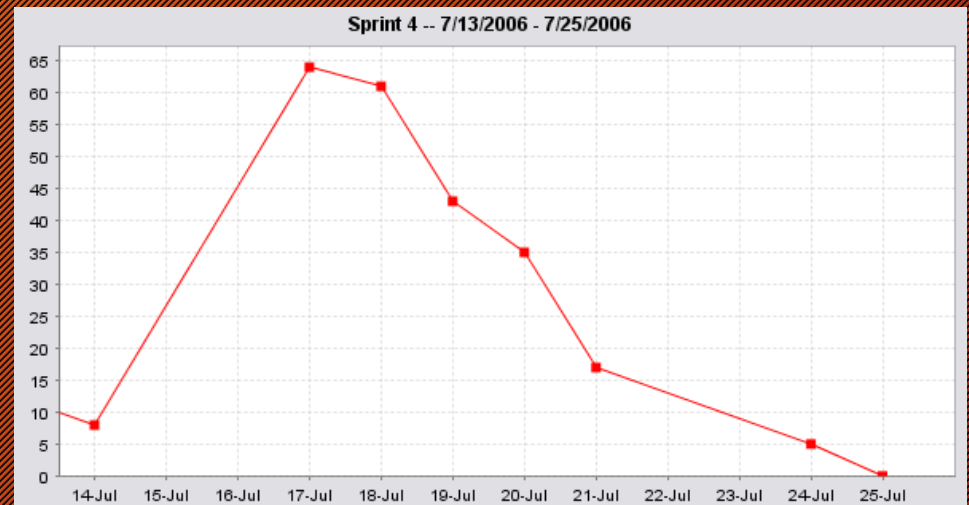


Sprint Backlog - another example

User Story	Tasks	Day 1	Day 2	Day 3	Day 4	Day 5	...
As a member, I can read profiles of other members so that I can find someone to date.	Code the ...	8	4	8	0		
	Design the ...	16	12	10	4		
	Meet with Mary about ...	8	16	16	11		
	Design the UI	12	6	0	0		
	Automate tests ...	4	4	1	0		
	Code the other ...	8	8	8	8		
As a member, I can update my billing information.	Update security tests	6	6	4	0		
	Design a solution to ...	12	6	0	0		
	Write test plan	8	8	4	0		
	Automate tests ...	12	12	10	6		
	Code the ...	8	8	8	4		

<http://www.mountangoatsoftware.com/scrum/sprint-backlog>

Sprint Burndown



<https://scrumology.com/sprint-burn-graph-signatures/>

Definition of Done*

- Verifying that your team's DoD meets these criteria will ensure that you are delivering features that are truly done, not only in terms of functionality but in terms of quality as well.
- Definition of Done is a simple list of activities (writing code, coding comments, unit testing, integration testing, release notes, design documents, etc.) that add verifiable/ demonstrable value to the product.

* not unique to Agile!

More Definition of Done

Joe, the Developer, waltzed into work one sunny Tuesday morning and was approached by Kenny, the Project Manager, asking if the feature Joe was working on was done. Joe had checked in his code to their shared source code repository yesterday afternoon and had unit tested it before doing so. With an emphatic “yes” Joe confirmed the feature’s completion. Kenny sighed in relief and said “great, then we will go ahead and deploy it into the UAT environment for our customer advocates to view”. Joe quickly backtracked on his original answer and blurted out “but it has not been fully tested by QA, the documentation is not updated, and I still need to pass a code review before it is finished”.

Done Checklist

1. Code produced (all 'to do' items in code completed)
2. Code commented, checked in and run against current version in source control
3. Peer reviewed (or produced with pair programming) and meeting development standards
4. Builds without errors
5. Unit tests written and passing
6. Deployed to system test environment and passed system tests
7. *Passed UAT (User Acceptance Testing) and signed off as meeting requirements* **For us: client has approved**
8. Any build/deployment/configuration changes implemented/documented/communicated **For us: product delivered**
9. Relevant documentation/diagrams produced and/or updated
10. Remaining hours for task set to zero and task closed **For us: final report**

Specifically for us

- Check with client, be prepared to tell your advisor:
 - How product will be delivered
 - Whether the clients want to do a code review
 - Clients standards/process for unit testing
 - Whether code needs to be integrated with existing system (will require help from developers)

Final Thought from the trenches

“Scrum is hard, but it sure is a whole lot better than what we were doing before!”