

An Introduction to Scrum

Filomena Ferrucci



Mountain Goat Software, LLC



1

We're losing the relay race

"The... 'relay race' approach to product development...may conflict with the goals of maximum speed and flexibility. Instead a holistic or 'rugby' approach—where a team tries to go the distance as a unit, passing the ball back and forth—may better serve today's competitive requirements."

Hiroataka Takeuchi and Ikujiro Nonaka, "The New Product Development Game", *Harvard Business Review*, January 1986.



Mountain Goat Software, LLC



2

Scrum in 100 words

- Scrum is an agile process that allows us to focus on delivering the highest business value in the shortest time.
- It allows us to rapidly and repeatedly inspect actual working software (every two weeks to one month).
- The business sets the priorities. Teams self-organize to determine the best way to deliver the highest priority features.
- Every two weeks to a month anyone can see real working software and decide to release it as is or continue to enhance it for another sprint.



Mountain Goat Software, LLC



3

Scrum origins

- Jeff Sutherland
 - Initial scrums at Easel Corp in 1993
 - IDX and 500+ people doing Scrum
- Ken Schwaber
 - ADM
 - Scrum presented at OOPSLA 95 with Sutherland
 - Author of three books on Scrum
- Mike Beedle
 - Scrum patterns in PLOPD4
- Ken Schwaber and Mike Cohn
 - Co-founded Scrum Alliance in 2002, initially within the Agile Alliance



Mountain Goat Software, LLC



4

Scrum has been used by:

- Microsoft
- Yahoo
- Google
- Electronic Arts
- High Moon Studios
- Lockheed Martin
- Philips
- Siemens
- Nokia
- Capital One
- BBC
- Intuit
- Intuit
- Nielsen Media
- First American Real Estate
- BMC Software
- Ipswitch
- John Deere
- Lexis Nexis
- Sabre
- Salesforce.com
- Time Warner
- Turner Broadcasting
- Oce



Mountain Goat Software, LLC



5

Scrum has been used for:

- Commercial software
- In-house development
- Contract development
- Fixed-price projects
- Financial applications
- ISO 9001-certified applications
- Embedded systems
- 24x7 systems with 99.999% uptime requirements
- the Joint Strike Fighter
- Video game development
- FDA-approved, life-critical systems
- Satellite-control software
- Websites
- Handheld software
- Mobile phones
- Network switching applications
- ISV applications
- Some of the largest applications in use



Mountain Goat Software, LLC



6

Characteristics

- Self-organizing teams
- Product progresses in a series of month-long “sprints”
- Requirements are captured as items in a list of “product backlog”
- No specific engineering practices prescribed
- Uses generative rules to create an agile environment for delivering projects
- One of the “agile processes”

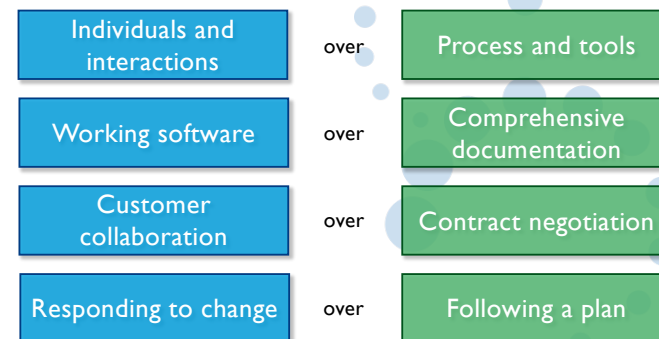


Mountain Goat Software, LLC



7

The Agile Manifesto—a statement of values

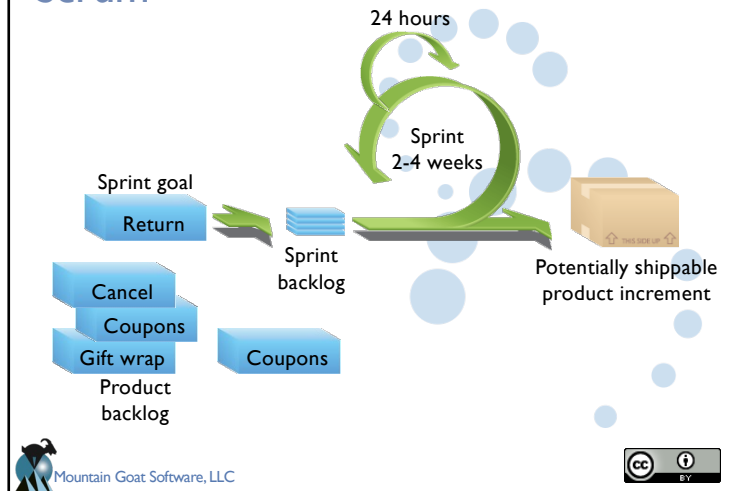
Source: www.agilemanifesto.org

Mountain Goat Software, LLC



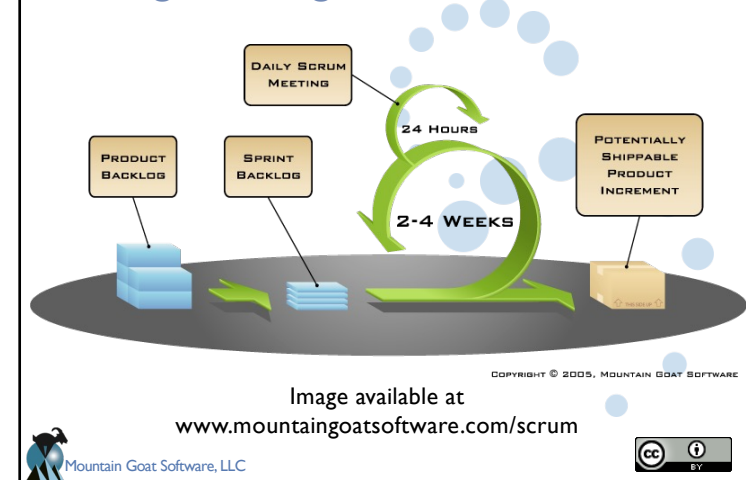
8

Scrum



9

Putting it all together



10

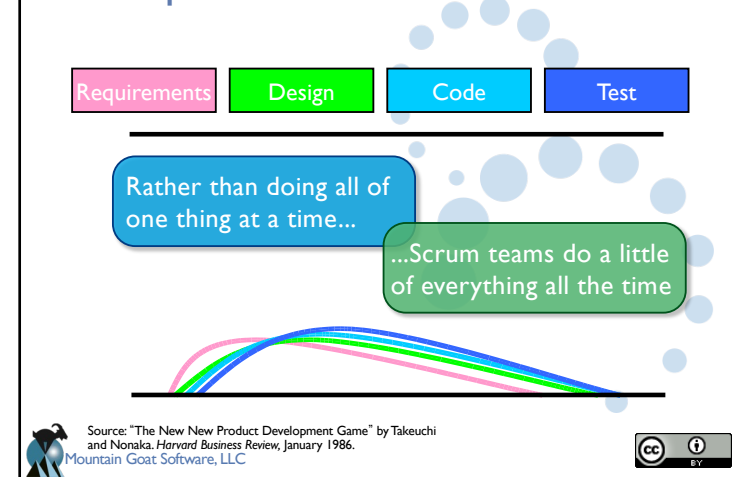
Sprints

- Scrum projects make progress in a series of “sprints”
- Analogous to Extreme Programming iterations
- Typical duration is 2–4 weeks or a calendar month at most
- A constant duration leads to a better rhythm
- Product is designed, coded, and tested during the sprint



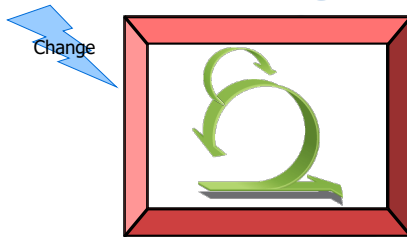
11

Sequential vs. overlapping development



12

No changes during a sprint



- Plan sprint durations around how long you can commit to keeping change out of the sprint



13

Scrum framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

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

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts



14

Scrum framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

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

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts



15

Product owner



- Define the features of the product
- Decide on release date and content
- Be responsible for the profitability of the product (ROI)
- Prioritize features according to market value
- Adjust features and priority every iteration, as needed
- Accept or reject work results



16

The ScrumMaster



- Represents management to the project
- Responsible for enacting Scrum values and practices
- Removes impediments
- Ensure that the team is fully functional and productive
- Enable close cooperation across all roles and functions
- Shield the team from external interferences



Mountain Goat Software, LLC



17

The team



- Typically 5-9 people
- Cross-functional:
 - Programmers, testers, user experience designers, etc.
- Members should be full-time
 - May be exceptions (e.g., database administrator)



Mountain Goat Software, LLC



18

The team



- Teams are self-organizing
 - Ideally, no titles but rarely a possibility
- Membership should change only between sprints



Mountain Goat Software, LLC



19

Scrum framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

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

Artifacts

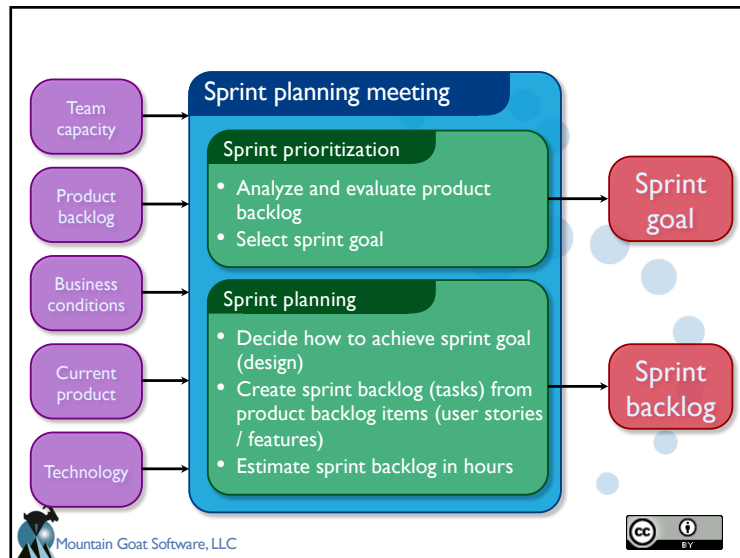
- Product backlog
- Sprint backlog
- Burndown charts



Mountain Goat Software, LLC



20



21

Team Capacity

- During sprint planning, teams face the challenge of sprint commitments.
 - How many stories can we commit in this sprint?
 - How to plan for the team capacity?
- Team capacity is calculated as per people availability in that sprint.
 - Ex. Team is of 5 people, then total capacity assuming 8 hour day, 2 weeks sprint(10 days) is = $5 \times 8 \times 10 = 400$ hours. NOOOOO!
 - Planning for this total capacity will be disaster. It will lead to team working over time, rushing towards the end, quality cuts and low team morale.

Mountain Goat Software, LLC

22

Team Capacity & Focus Factor

Focus Factor (F.F)

- is teams ability to remain focussed on the sprint goals without any other distractions.
- lies in the range **0.6 – 0.8**
- After multiplying total capacity with focus factor you get real capacity against which you can make sprint commitments or forecasting. This is the effective hours you can expect from the team.
- applying focus factor say 0.6, then this team real capacity will be $400 \times 0.6 = 240$ hours
- Team will take on the stories till the time all the tasks sum to not more than 240 hours(in this example).

Mountain Goat Software, LLC

23

Team Capacity & Focus Factor

- Use lesser **focus factor** on the following situations:
 - When team is **starting new on a project**
 - Team is using **scrum for the first time**
 - Team is working on a **complex product or new to technology domain**
 - Team is **less matured, needs lot of handholding ...**
 - people being allocated to multiple projects, overhead of task switching comes to play**

Tips: start a team on successful note(improves morale). Using lesser focus factor when you start fresh and then if team meets sprint goals early, then they can take up more in the current sprint. Retrospect on this in coming sprints to see if you want to **increase focus factor marginally and fine tune**, iterate this factor as you go, to reach sustainable pace/Flow. Going beyond 0.8 **can be risky** and can derail teams too.

Mountain Goat Software, LLC

24

Team Capacity & Focus Factor

- organisation or product development is very chaotic then this factor will remain on extreme left like 0.6 or may be below.
- **Chaotic organisations** have lot of **unplanned meetings**, pre-sales urgency, hiring team coming to the project team at a **last minute with a interview request**, not having defined core working hours, lesser clarity sprint backlog, wrong team structures(read too much inter-dependency) and list goes on... To summarise – **no rhythm**.
- **Tips:** start a team on successful note(improves morale). Using lesser focus factor when you start fresh and then if team meets sprint goals early, then they can take up more in the current sprint. Retrospect on this in coming sprints to see if you want to **increase focus factor marginally and fine tune**, iterate this factor as you go, to reach sustainable pace/Flow. Going beyond 0.8 can be risky and can derail teams too.



Mountain Goat Software, LLC



25

Team Capacity vs Velocity

- **What is Team's velocity**
Number of **story points** delivered/demo in a Sprint is called velocity. For example, if team planned 30 story point (Business value) worth of user stories in a sprint and able to deliver as planned then team's velocity is 30.



Mountain Goat Software, LLC



26

Story points

- Traditional software teams give estimates in a time format: days, weeks, months. Many agile teams use story points.
- Story points are units of measure for expressing an estimate of the overall effort required to fully implement a product backlog item (e.g., a user story).
- In simple terms, a story point is a number that tells the team about the *difficulty level of the story*. Difficulty could be related to complexities, risks, and efforts involved.
- Story point estimation, a kind of **relative estimation**: In order to do that each team would have to find a baseline story. It does not necessarily to be the smallest one, but the one that everyone within the team can resonate with. Once determined, sizing of all the user stories should be initiated by comparing them against the baseline.
- When estimating new stories all you have to do is pick a story and say: "will this take longer than reference story x?" or "will it be less than reference y?" Teams assign story points relative to work complexity, the amount of work, and risk or uncertainty.

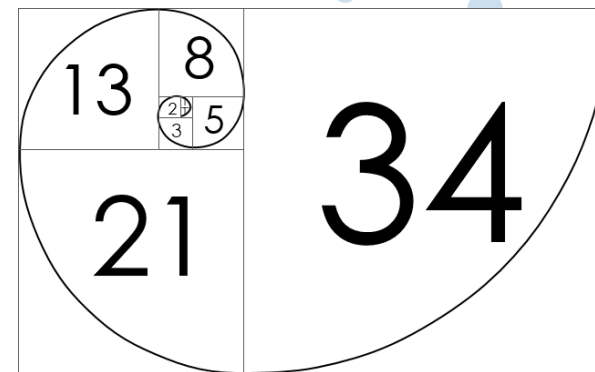


Mountain Goat Software, LLC



27

Fibonacci number for Story Point



Mountain Goat Software, LLC



28

Story points vs. hours: few reasons to use story points

- Dates don't account for the non-project related work that inevitably creeps into our days: emails, meetings, and interviews that a team member may be involved in.
- Dates have an emotional attachment to them. Relative estimation removes the emotional attachment.
- Once you agree on the relative effort of each story point value, you can assign points quickly without much debate.
- Story points reward team members for solving problems based on difficulty, not time spent. This keeps team members focused on shipping value, not spending time.



29

Story points vs. hours

- Unfortunately, story points are often misused.
- Story points go wrong when they're used to judge people, assign detailed timelines and resources, and when they're mistaken for a measure of productivity.
- Instead, teams should use story points to understand the size of the work and the prioritization of the work.



30

Story points and planning poker

- The team will take an item from the backlog, discuss it briefly, and each member will mentally formulate an estimate.
- Then everyone holds up a card with the number that reflects their estimate.
- If everyone is in agreement, great! If not, take some time (but not too much time—just couple minutes) to understand the rationale behind different estimates.
- Remember though, estimation should be a high level activity. If the team is too far into the weeds, take a breath, and up-level the discussion.



31

How to measure velocity

- Take the amount of story points your team completed in three past sprints, add them together, and divide by three (the amount of sprints). That average is your basic velocity. The more sprints you add to your velocity measurement, the more accurate your average.
- With that data point in mind, you can then extrapolate how many story points your team can complete in the next sprint.
- Velocity is based on basic math and basic inference. Once you have the average, you can predict that that's the amount you'll likely do in the next sprint, too. Barring extreme circumstances, your velocity will give you a good baseline for predicting how many points you'll complete in the next sprint.



32

Capacity planning Strategies

- Capacity is about estimation, but the estimate is educated.
- Capacity uses velocity, an average, as a starting point. From that baseline, you can build a prediction of how much you can do over the next sprint informed by more immediate circumstances over a time frame.
- Estimate capacity by quantifying the amount of engineering time each team member can work in the upcoming sprint after accounting for time off, potential illness, and responsibilities outside of story development (such as maintenance work, PR review, meetings, etc.).
- To use them in tandem, find your velocity from your past sprints, and then adjust that average based on expected capacity from your current vantage point. Any given sprint might be above or below average; the key is using the information you have to make as accurate a prediction as possible.



33

Team Capacity vs Velocity

- Capacity planning is an **estimate** of the future; velocity is a **measurement** of the past.
- Capacity is an *estimate* of the *total* amount of engineering time available in a given sprint. Effective capacity planning should be based on future expectations of available time, i.e., an estimate relative to the expected future.
- Velocity is a *measurement* of the *average* amount of story points delivered in a given time period. Planning based on velocity means basing your estimates on past performance, i.e., an estimate relative to the measured past.



34

User Stories

- very slim and high-level requirements artifacts (functional, non functional, constraints...)
- Students can purchase monthly parking passes online.
- Parking passes can be paid via credit cards.
- Parking passes can be paid via PayPal.
- Professors can input student marks.
- Students can obtain their current seminar schedule.
- Students can order official transcripts.
- Students can only enroll in seminars for which they have prerequisites.
- Transcripts will be available online **via a standard browser**.



35

User Stories

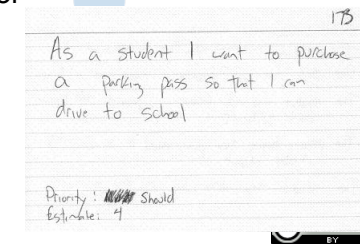
Format:

As a (role) I want (something) so that (benefit)

"As a Student I want to purchase a parking pass so that I can drive to school"

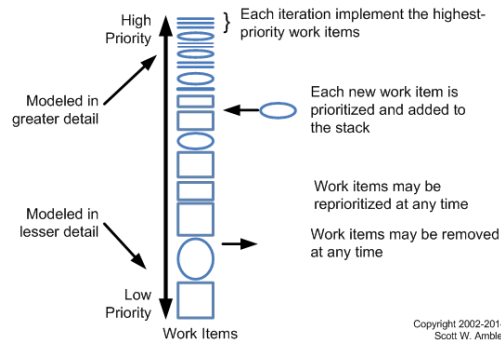
User story card
(stakeholder)

Epics & Themes



36

Disciplined agile change management process



Copyright 2002-2014
Scott W. Ambler



37

Sprint planning

- Team selects items from the product backlog they can commit to completing
- Sprint backlog is created
 - Tasks are identified and each is estimated (1-16 hours)
 - Collaboratively, not done alone by the ScrumMaster
- High-level design is considered

As a vacation planner, I want to see photos of the hotels.

Code the middle tier (8 hours)
Code the user interface (4)
Write test fixtures (4)
Code the foo class (6)
Update performance tests (4)



38

The daily scrum

- Parameters
 - Daily
 - 15-minutes
 - Stand-up
- Not for problem solving
 - Whole world is invited
 - Only team members, ScrumMaster, product owner, can talk
- Helps avoid other unnecessary meetings



39

Everyone answers 3 questions

1
What did you do yesterday?

2
What will you do today?

3
Is anything in your way?

- These are **not** status for the ScrumMaster
 - They are **commitments** in front of peers



40

The sprint review

- Team presents what it accomplished during the sprint
- Typically takes the form of a demo of new features or underlying architecture
- Informal
 - 2-hour prep time rule
 - No slides
- Whole team participates
- Invite the world



Mountain Goat Software, LLC



41

Sprint retrospective

- Periodically take a look at what is and is not working
- Typically 15–30 minutes
- Done after every sprint
- Whole team participates
 - ScrumMaster
 - Product owner
 - Team
 - Possibly customers and others



Mountain Goat Software, LLC



42

Start / Stop / Continue

- Whole team gathers and discusses what they'd like to:

Start doing

Stop doing

Continue doing

This is just one of many ways to do a sprint retrospective.



Mountain Goat Software, LLC



43

Scrum framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

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

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

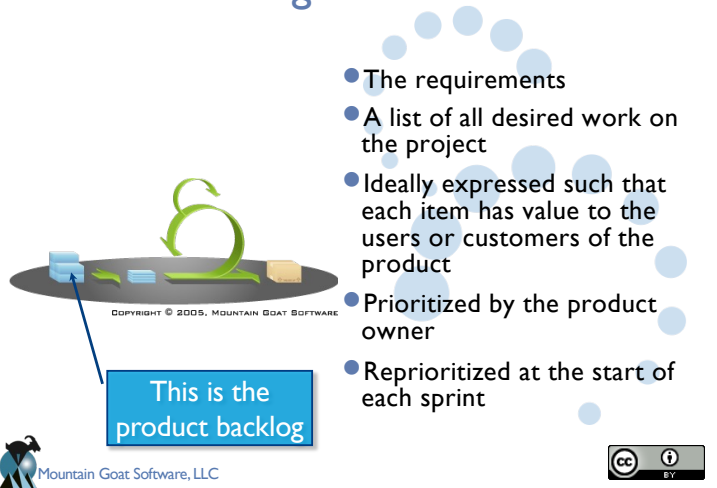


Mountain Goat Software, LLC



44

Product backlog



Copyright © 2005, Mountain Goat Software

This is the product backlog

- The requirements
- A list of all desired work on the project
- Ideally expressed such that each item has value to the users or customers of the product
- Prioritized by the product owner
- Reprioritized at the start of each sprint

Mountain Goat Software, LLC

45

A sample product backlog

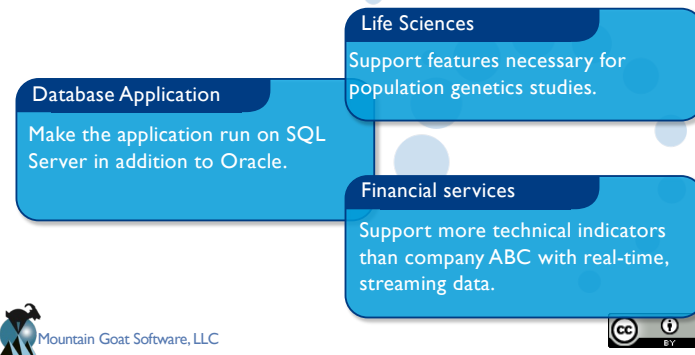
Backlog item	Estimate
Allow a guest to make a reservation	3
As a guest, I want to cancel a reservation.	5
As a guest, I want to change the dates of a reservation.	3
As a hotel employee, I can run RevPAR reports (revenue-per-available-room)	8
Improve exception handling	8
...	30
...	50

Mountain Goat Software, LLC

46

The sprint goal

- A short statement of what the work will be focused on during the sprint



Life Sciences
Support features necessary for population genetics studies.

Database Application
Make the application run on SQL Server in addition to Oracle.

Financial services
Support more technical indicators than company ABC with real-time, streaming data.

Mountain Goat Software, LLC

47

Managing the sprint backlog

- Individuals sign up for work of their own choosing
 - Work is never assigned
- Estimated work remaining is updated daily

Mountain Goat Software, LLC

48

Managing the sprint backlog

- Any team member can add, delete or change the sprint backlog
- Work for the sprint emerges
- If work is unclear, define a sprint backlog item with a larger amount of time and break it down later
- Update work remaining as more becomes known



Mountain Goat Software, LLC



49

A sprint backlog

Tasks	Mon	Tues	Wed	Thur	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	4	
Test the middle tier	8	16	16	11	8
Write online help	12				
Write the foo class	8	8	8	8	8
Add error logging			8	4	



Mountain Goat Software, LLC



50

A sprint backlog with user stories

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		

During the Scrum sprint, team members are expected to update the sprint backlog as new information is available, but minimally once per day. Many teams will do this during the daily scrum. Once each day, the estimated work remaining in the sprint is calculated and graphed by the ScrumMaster, resulting in a sprint burndown chart like this one.



51

Sprint Burndown Chart

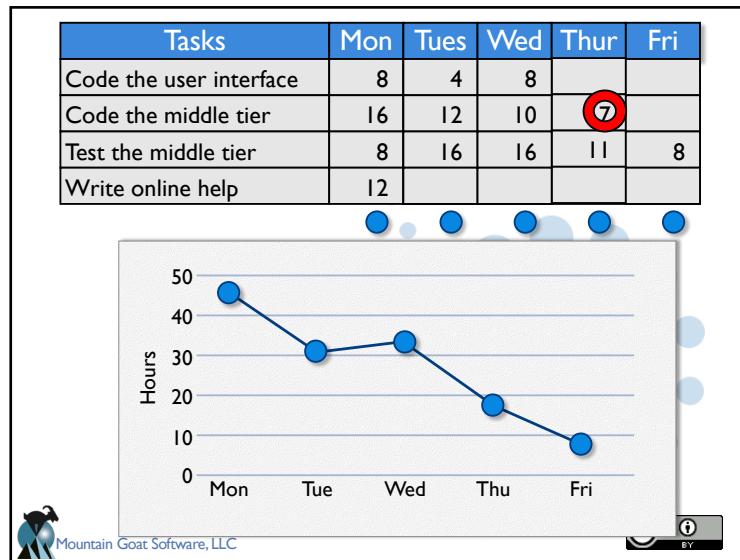
- Teams use the sprint Burndown chart to track the product development *effort remaining in a sprint*.
- It should consist of:
 - X axis to display working days
 - Y axis to display remaining effort
 - Ideal effort as a guideline
 - Real progress of effort



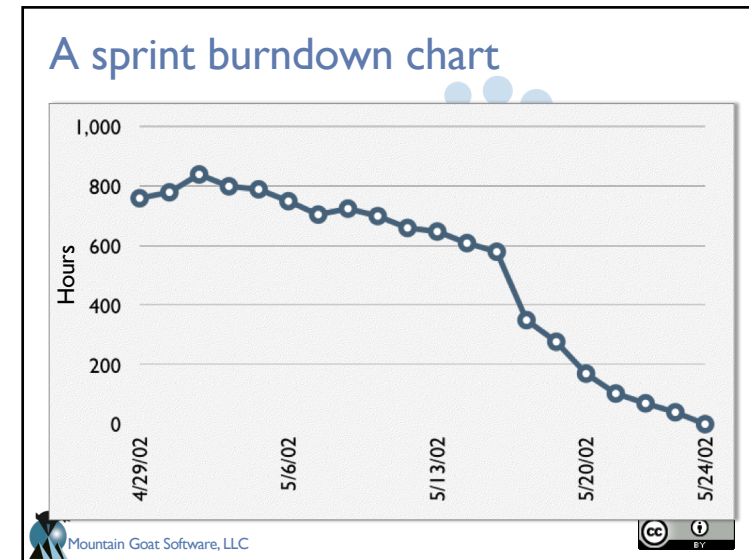
Mountain Goat Software, LLC



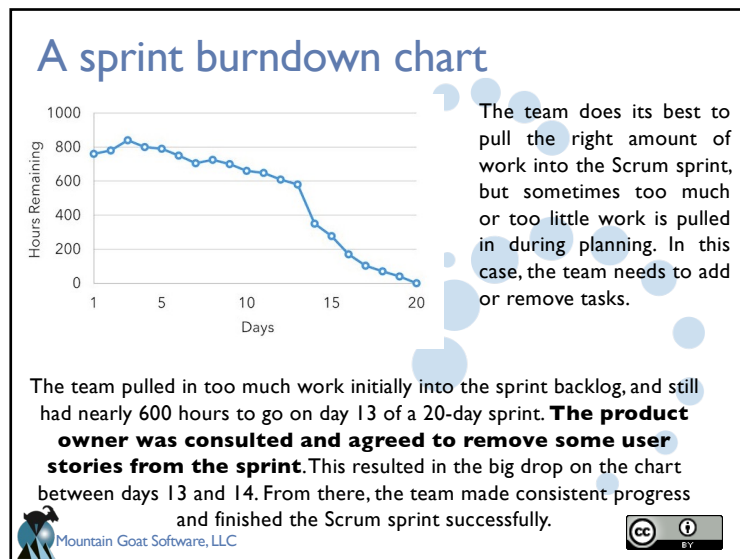
52



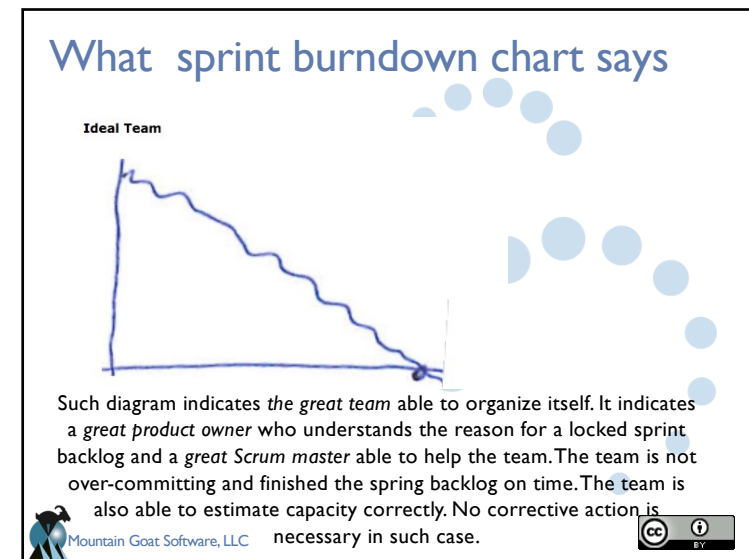
53



54



55



56

What sprint burndown chart says

Great Team



Experienced teams. The team has completed work on time and met the sprint goal. They also have applied the principle of getting things done, but the most important is they have *adapted a scope of the sprint backlog* to complete the sprint. At the end the team has a possibility to complete some additional work.

In the retrospective, the team should discuss the reasons of late progress in the first half of the sprint and solve issues so they are better in the next sprint. The team should also consider the **capacity** that they are able to complete.



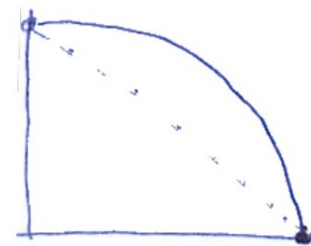
Mountain Goat Software, LLC



57

What sprint burndown chart says

Nice Team



This is a *typical progress* that can be observed in many experienced agile teams.

The chart says again that the team was able to complete their commitment on time. They adapted the scope or worked harder to complete the sprint. The team is self-reflecting.

The team should discuss change of plan immediately as they see the progress has been slowing down from the beginning of the sprint. Typically it is suggested to move a low priority item from the sprint backlog to the next sprint or back to the product backlog.



Mountain Goat Software, LLC



58

What sprint burndown chart says

Boom. It Is Too Late.



This burndown chart says: "You have not completed your commitment".

The team has been late for the entire sprint. The team did not adapt the sprint scope to appropriate level. It shows that the team has not completed stories that should have been split or moved to the next sprint.

In such situation the capacity of the next sprint should be lowered. If this happens again, corrective actions should be taken after a few days when slower progress is observed. Typically, lower priority story should be moved to the next sprint or back to the product backlog.



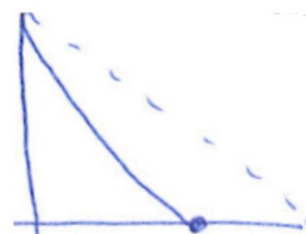
Mountain Goat Software, LLC



59

What sprint burndown chart says

Boom. Too Early.



The team finishes its work sooner than expected. The stories were implemented, but the team didn't work on additional stories even it had the capacity to do it.

The stories were probably *overestimated*, therefore the team finished them earlier. Also the velocity of the team has not been probably estimated correctly.

The *Scrum Master* must be more proactive in either getting the team to fix estimation or ensure additional stories are ready to be added into the current sprint.



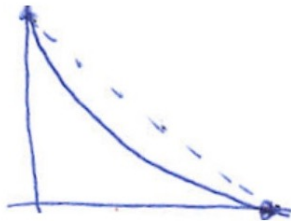
Mountain Goat Software, LLC



60

What sprint burndown chart says

Let's Have a Rest



The team with such progress has a problem. The problem is either the team committed to less than they are able to complete or the product owner does not provide enough stories for the sprint.

The reason might be also an over-estimation of complexity, which ends up in completion earlier than expected at the beginning of the sprint.

The Scrum Master should identify this problem earlier and ask the product owner to provide the team with more work. Even if stories are over-estimated, the team should at least continue with stories from the next, already preplanned, sprint.



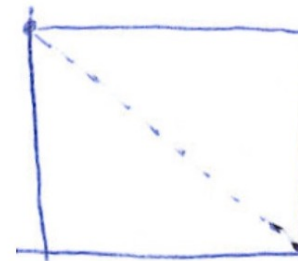
Mountain Goat Software, LLC



61

What sprint burndown chart says

Oh, Management Is Coming!



- The team is probably doing some work, but maybe it does not update its progress accordingly.
- Another reason might be that the product owner has added the same amount of work that was already completed, therefore the line is straight.
- The team is not able to predict the end of the sprint or even to provide the status of the current sprint.

The Scrum Master should improve its Scrum masterships and coach the team on why it is necessary to track the progress and how to track it. Such team should be stopped after two or three days that shows a flat line of progress and should immediately apply corrective actions.



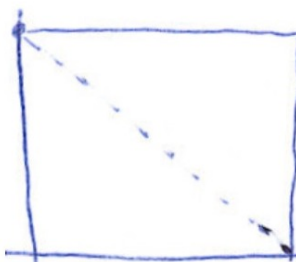
Mountain Goat Software, LLC



62

What sprint burndown chart says

Oh, Management Is Coming!



- The team is non-functional on many levels.
- The Scrum Master of this team is not able to coach the team why it is necessary to track progress on daily basis.
- The product owner does not care about development progress either.

To fix this situation the team should restart. *Restart from scratch* by training and do a retrospective to figure out why this is happening.



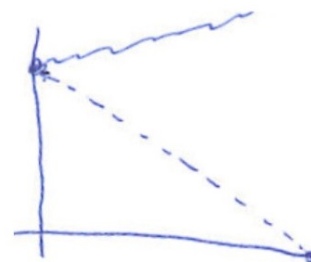
Mountain Goat Software, LLC



63

What sprint burndown chart says

Up to the Sky



The first sprint typically looks like that.

Stories or tasks were added into the sprint backlog everyday without any progress recorded.

Another reason might be that tasks were re-estimated constantly during the sprint.

The mistake is that the team did not identify the problem: The sprint backlog should be reevaluated and rearranged immediately. The coach might be helpful, as an experienced Scrum master and product owner should often facilitate this situation.



Mountain Goat Software, LLC



64

What sprint burndown chart says

Progress from Long-Run Perspective

- Does a nice Burndown chart indicate a *great* team?
 - Maybe if your team indicates *great* progress for more than one iteration.
- Does the team believe in such success?
 - Be careful! We all know about changes coming every minute. Maybe the team provides conservative estimation for their safety!
- Management usually takes care about the improvement of velocity, sprint by sprint. Please, do not expect that.
- Velocity is not an indicator of the team. *Velocity is not a KPI* by which you should measure your team. *Velocity is just capacity planning tool.* Nothing more, nothing less.
- Asking people to accomplish more story points in iterations will result in stories that have more story points estimated without real reasons. It could be name as "*Story points inflation*".



Mountain Goat Software, LLC



65

Scalability

- Typical individual team is 7 ± 2 people
 - Scalability comes from teams of teams
- Factors in scaling
 - Type of application
 - Team size
 - Team dispersion
 - Project duration
- Scrum has been used on multiple 500+ person projects

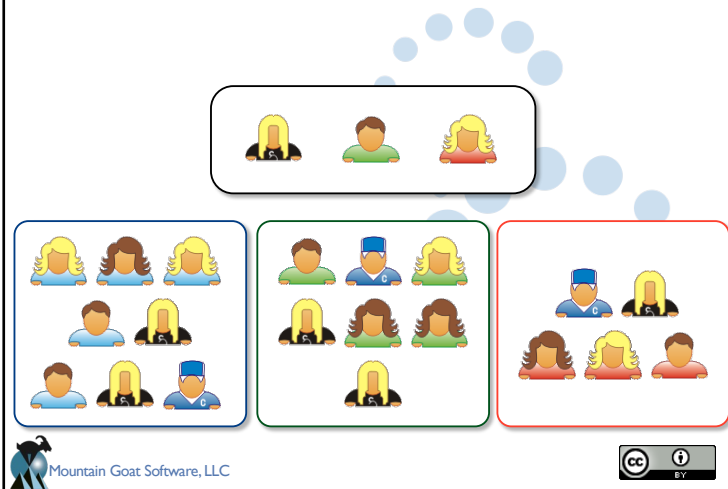


Mountain Goat Software, LLC



66

Scaling through the Scrum of scrums

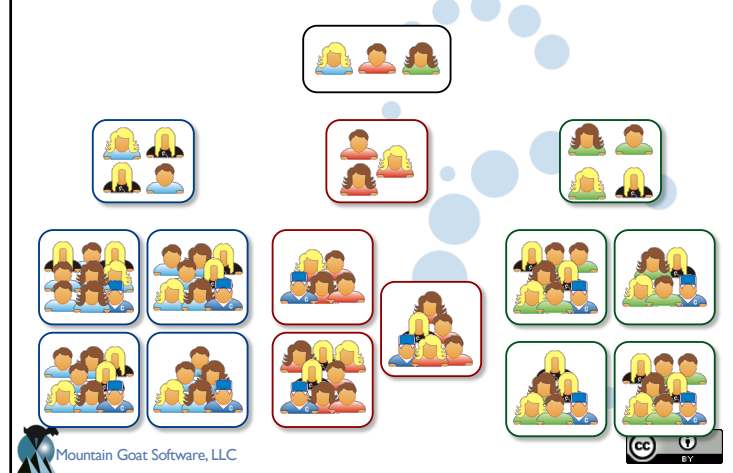


Mountain Goat Software, LLC



67

Scrum of scrums of scrums



Mountain Goat Software, LLC



68

Where to go next

- www.mountaingoatsoftware.com/scrum
- www.scrumalliance.org
- www.controlchaos.com
- scrumdevelopment@yahoogroups.com



69

A Scrum reading list

- *Agile and Iterative Development: A Manager's Guide* by Craig Larman
- *Agile Estimating and Planning* by Mike Cohn
- *Agile Project Management with Scrum* by Ken Schwaber
- *Agile Retrospectives* by Esther Derby and Diana Larsen



70

A Scrum reading list

- *Agile Software Development Ecosystems* by Jim Highsmith
- *Agile Software Development with Scrum* by Ken Schwaber and Mike Beedle
- *Scrum and The Enterprise* by Ken Schwaber
- *Succeeding with Agile* by Mike Cohn
- *User Stories Applied for Agile Software Development* by Mike Cohn



71

Copyright notice



- You are free:
 - to Share—to copy, distribute and and transmit the work
 - to Remix—to adapt the work
- Under the following conditions
 - Attribution. You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
- Nothing in this license impairs or restricts the author's moral rights.
- For more information see <http://creativecommons.org/licenses/by/3.0/>



72

Contact information



Presentation by: Mike Cohn
 mike@mountaingoatsoftware.com
 www.mountaingoatsoftware.com
 (720) 890-6110 (office)

You can remove this (or any slide) but you must credit the source somewhere in your presentation. Use the logo and company name (as at bottom left, for example) or include a slide somewhere saying that portions (or all) of your presentation are from this source. Thanks.



Mountain Goat Software, LLC

