Agile Estimating and Planning
Special thanks to

Mike Cohn,
founder of
Mountain Goat Software, who gave us permission to use this material.

mountaingoatsoftware.com
720-890-6110
Where’s my plan?
Why plans go wrong...
Lateness passed down schedule

- Task 3 late when?
Have you ever started a term paper the night before it was due?
Student syndrome

• It is based on estimates like this:

Task

Local Safety
• It is based on estimates like this:

- Task
- Local Safety

• But we do this:

- Local Safety
- Task
Multitasking

- 20-30% time increase to do each task
- Greater loss on complex tasks
To do two things at once is to do neither.

- Publilius Syrus
The Effects of Multitasking

![Graph showing the relationship between the number of assigned tasks and the percent of time spent on value-adding tasks. The graph indicates a downward trend as the number of assigned tasks increases.]
What’s a good plan?
“It is a bad plan that admits to no modifications.”

-- Publilius Syrus (ca. 42 BCE)
What’s a Good Plan?

- Supports reliable decision-making

Go from:

- Will be done in fourth quarter
- ... in November
- .... November 7
A Plan is NOT a Commitment

It is a current view.

If one sticks to the idea that once set, a plan should not be changed, a business cannot exist for long.

--- Taiichi Ohno

In preparing for battle I have always found that plans are useless, but planning is indispensable.

--- Dwight D. Eisenhower
A Plan is NOT a Commitment

- If plans are commitments, then we are committing to decisions made **when we were the most ignorant** (recall the cone of uncertainty, 5%)

- Measuring conformance to plan is measuring the wrong thing because the plan will change
What makes planning Agile?

- More focused on planning than the plan
- Encourages change
- Plans are easily changed
- Done throughout the project
What’s an Agile plan?

In the form of three backlogs:

- **Product** Backlog
  Epics and Themes for Product

- **Release** Backlog
  Release Theme and User Stories

- **Sprint** Backlog
  User Stories and tasks planned for the Sprint (iteration)
Product Planning
Product Planning

✔ Product Backlog:
  - Develop Epic User Stories
  - Prioritize based on Business Value
  - Define release themes
  - Place Epics into releases

Who: Stakeholders, Business and Team
Release Planning
Release Planning

Product Backlog (Prioritized Epics and Themes)

Release Planning Meeting

Release Backlog for Epic

High BV  |  Medium BV  |  Low BV

Priorities: H, M, L
Release Backlog:

- Develop User Stories for ONE release
- Prioritize based on Business Value

Who: **Stakeholders, Business and Team**
Release Planning, part 2

- Estimate Story Points on User Stories

Who: Delivery Team
Agile Estimating
Exercise: How long will it take ...

.... to **read** the latest Harry Potter book?

.... to **drive** to Austin, TX?
Estimate Size by Deriving Duration

Size → Calculation → Duration

300 kilograms → Velocity = 20 → 300/20 = 15 iterations
measure of size
Traditional Measure of Size

Traditional measures of size:
- Lines of Code
- Function Points
Agile Measure of Size

[0.5] As a loan rep I want to see all pending loans in my pipeline

Agile measures of size:

Story Points
Story Points

- The “bigness” of a task
- Points are unit-less
- Influenced by
  - How hard it is
  - How much there is

As a buyer, I want to be able to have some but not all items in my cart gift wrapped. 8
Story Points

- Relative values are what is important:
  - A login screen is a 2.
  - A search feature is an 8.
- Basic math properties hold, e.g., 5+5 = 10

As a mailer, I want to produce discounted mailings for 1st class automation letters and flats 30
Exercise: Dog Points

Assign “dog points” to the following breeds:

- Labrador Retriever
- Dachshund
- Great Dane
- Terrier
- German Shepherd
- Poodle
- St. Bernard
- Bulldog
Why Story Points

• Help drive cross-functional behavior
• Estimates do not decay
• Pure measure of size
• Estimating is typically faster
Estimating Tips and Techniques
Estimate by Analogy

- Compare one user story to another
  - “This story is like that story”
- Don’t use a single gold standard
  - Triangulate: Compare the story to be estimated to other estimated stories
### Triangulation

- Compare a story to similar stories
- Group like-sized stories

<table>
<thead>
<tr>
<th>5 pts</th>
<th>3 pts</th>
<th>2 pts</th>
<th>1 pt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Story A</td>
<td>Story D</td>
<td>Story E</td>
<td>Story H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Story J</td>
<td>Story K</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Story C</td>
<td>Story F</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Story B</td>
<td>Story G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Story I</td>
</tr>
</tbody>
</table>
Disaggregation

- Breaking a big story into smaller stories
  - Can’t estimate the big story – don’t know enough
  - Break into smaller estimatable stories

- Don’t get too small
  - Small errors can add up
  - Tasks slip through cracks
How much effort?

- A little effort helps a lot
- A lot more effort only helps a little
Use the Right Units

- Can you distinguish a 1-point story from a 2?
- Can you distinguish a 17 from an 18?
- Use units that makes since, such as
  - 1, 2, 3, 5, 8, 13
  - 1, 2, 4, 8

Use Planning Poker Cards
Planning Poker

An iterative approach to planning:

- Team members use planning poker cards to make an estimate
- Product owner reads and discusses a story
- Each team member selects a card that’s her estimate, placing it face down
- When all cards are face down, turn them over
- Outliers are discussed
- Continue estimating and discussing until consensus reached
## Planning Poker Example

<table>
<thead>
<tr>
<th>Estimator</th>
<th>Round 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Susan</td>
<td>3</td>
</tr>
<tr>
<td>Vadim</td>
<td>8</td>
</tr>
<tr>
<td>Ann</td>
<td>2</td>
</tr>
<tr>
<td>Chris</td>
<td>5</td>
</tr>
</tbody>
</table>
**Planning Poker Example**

<table>
<thead>
<tr>
<th>Estimator</th>
<th>Round 1</th>
<th>Round 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Susan</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Vadim</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>Ann</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Chris</td>
<td>5</td>
<td>8</td>
</tr>
</tbody>
</table>
# Planning Poker Example

<table>
<thead>
<tr>
<th>Estimator</th>
<th>Round 1</th>
<th>Round 2</th>
<th>Round 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Susan</td>
<td>3</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Vadim</td>
<td>8</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Ann</td>
<td>2</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Chris</td>
<td>5</td>
<td>8</td>
<td>5</td>
</tr>
</tbody>
</table>
# Exercise: Agile Estimating

<table>
<thead>
<tr>
<th>Item #</th>
<th>Product Backlog Item</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Read a high-level 10-page overview of agile software development in <em>People</em> magazine.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Read a densely written 5-page research paper about agile software development in an academic journal.</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Write the product backlog for a simple eCommerce site that sells only clocks.</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Recruit, interview, and hire a new member for your team.</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Create a 60-minute presentation about agile estimating and planning for your co-workers.</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Wash and wax your boss’ Porsche.</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Read a 150-page book on agile software development.</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Write an 8-page summary of that book for your boss.</td>
<td></td>
</tr>
</tbody>
</table>
Planning Poker Guidelines

- Only **delivery team members** estimate their user stories
- Outliers explain their estimates
- Everyone’s opinion is heard
- It’s a **conversation**! Not a commitment.
Estimate Story Points on User Stories

Who: Delivery Team

Exercise: Estimate your user stories using story points.
Sprint Planning
Determine user stories in sprint
Define “done, done, done”
Team commits
Create an information radiator

Who: Delivery Team
Sprint Duration

How long can you keep business changes out of the sprint?
Sprint Planning Session

- Held at the beginning of a new Sprint
- Chaired by the Scrum Master
- Attended by all including Key Stakeholders
- Update the Product Backlog with new user stories
- Select highest priority items in the backlog based on Business Value And Optimization of team resources
Team may break down stories into tasks (all tasks must be done to demo User Story)

As a card holder, I want to be able to withdraw funds from my account so I can have cash available to me

- Make a withdrawal with sufficient funds
- Make a withdrawal with insufficient funds (exception)
- Make a withdrawal when specified demonization not available (exception)
Team Defines ‘Done3’

Consider:

- No showstoppers
- All errors that the team has not agreed to are removed
- Code is unit tested, function tested, system tested, performance tested, tested end-to-end
- A meaningful stakeholder review has been conducted
Can ‘Done3’ really be done?

This puts a high premium on:

- Valuable, maintained, nested automation
- Appropriate coverage (e.g. 80%)
- True test-driven development
- Avoiding technical debt
- Continuous integration
- Really understanding what quality looks like
Jeff Sutherland (co-creator of Scrum), said while @ PatientKeeper:

“It took us four years to get done, done, done, done.”

Patientkeeper provides safety critical patient management software
Example: ‘Done4’

What does “Done”, “Done”, “Done”, “Done” mean?

- It is fully developed
- It is fully tested
- It has no Severity 1s or 2s
- It has been deployed before a release in a client environment
Example: ‘Done4’

They ship

- A major release every 3 months
- A minor release every month
- And minor updates once a week

Consider the competitors, teams of 600-700 developers and they cannot achieve the work flow Patientkeeper does.
Information Radiator
Information Radiator

- Visual representation of progress
- Display of:
  - Work Planned (Product, Release and Sprint)
  - Work in Progress
  - Work Done
  - User Stories to mitigate risk
  - User Stories to gather information to make future decisions
## Sprint 2

<table>
<thead>
<tr>
<th>WP</th>
<th>WIP</th>
<th>WD</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="WP" /></td>
<td><img src="image2" alt="WIP" /></td>
<td><img src="image3" alt="WD" /></td>
</tr>
</tbody>
</table>

**New User Stories**

- ![New User Stories](image4)

**Release Backlog**

**Epic:** ![Epic](image5)
<table>
<thead>
<tr>
<th>WP</th>
<th>WIP</th>
<th>WD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BH</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CO</td>
<td></td>
</tr>
</tbody>
</table>

**New User Stories**

**Release Backlog**

**Epic:**
## Sprint 2

<table>
<thead>
<tr>
<th>WP</th>
<th>WIP</th>
<th>WD</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Images" /></td>
<td><img src="image2.png" alt="Images" /></td>
<td><img src="image3.png" alt="Images" /></td>
</tr>
</tbody>
</table>

- **New User Stories**

- **Release Backlog**
  - Epic: [Images](image4.png)

---

- **Information Radiator**
- **Release Backlog**
- **Epic:** [Images](image5.png)
Example: Information Radiator
Sprint Burndown Chart

Day Velocity: 5  
Forecasted Iteration End Date: 17-Jan-09

- effort todo yesterday
- completed today
- iteration ideal line
- forecasted progress (we are doing good)
- effort removed today (story deleted)
Release Burn Up Chart

Done for sprint results against total release backlog.
Exercise: Information Radiator

Create an information radiator

- Place all sprint stories onto the Information Radiator under Work Planned column
- Add a section for “New User Stories”
- Decide what ‘Done’ is
Velocity
Velocity

- **Long-term** measure of work completed in iterations
- A **GUIDE** not a goal.
Track Velocity Multiple Ways

Last Observation=36
Mean (last 8)=33
Mean (lowest 3)=28
Extrapolate from Velocity

Assume five iterations left

Finish here at **lowest** velocity: 5x28

Finish here at **average** velocity: 5x33

Finish here at **current** velocity: 5x36
How much can I get by <date>?
Fixed Date ‘Planning’

- Determine how many iterations you have
- Estimate high and low velocity
- Low velocity times iterations = ‘will have’ stories
- High velocity times iterations = ‘might have’
- All the rest = ‘will not have’
## Fixed Date ‘Planning’ Example

<table>
<thead>
<tr>
<th>Desired Release Date</th>
<th>30 June</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Date</td>
<td>1 January</td>
</tr>
<tr>
<td>Number of Iterations</td>
<td>12 (two weeks)</td>
</tr>
<tr>
<td>Low Velocity</td>
<td>15</td>
</tr>
<tr>
<td>High Velocity</td>
<td>20</td>
</tr>
</tbody>
</table>

- **Will have**
  - 12x15
- **Might have**
  - 12x20
- **Won’t have**
Agile Estimating and Planning Summary
# Product Planning

## Product Backlog

<table>
<thead>
<tr>
<th>Input</th>
<th>SOW, Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>Prioritized Product Backlog (Epics and Release Themes)</td>
</tr>
<tr>
<td>Who</td>
<td>Business, Product Owner, Dev Team representatives</td>
</tr>
</tbody>
</table>
Release Planning, part 1

Product Backlog

Release Backlog

Theme

Input | Product Backlog
--- | ---
Output | Prioritized Release Backlog (User Stories) marked as differentiating or parity
Who | Business, Product Owner, Dev Team
## Release Planning, part 2

### Input
- Release Backlog

### Output
- Estimated and Prioritized Release Backlog (story points)

### Who
- Delivery Team using planning poker
Release Planning, part 2

Input: Release Backlog

Output: Estimated and Prioritized Release Backlog (story points)

Who: Delivery Team using planning poker
### Sprint Planning

<table>
<thead>
<tr>
<th>WP</th>
<th>WIP</th>
<th>WD</th>
</tr>
</thead>
</table>

#### Input
- Release Backlog

#### Output
- Sprint Backlog, Definition of Done, Architecture Spike, Information Radiator

#### Who
- Delivery Team

#### New User Stories
Refer to Mike Cohn’s book for details on how to estimate and plan.
Scrum Deep Dive

Questions

Estimating & Planning
For every sprint ...
Scrum

- Release Backlog
- Product Backlog
- Sprint Backlog
- Daily Scrum Meeting

24 Hours

2-4 Weeks

Potentially Shippable Product Increment
Daily Standup Scrum Meetings

- Daily 15 minute status meeting
- Same place and time every day
- Chaired by Scrum Master
- Attended by entire sprint team
- Others can attend
- Chickens and pigs (only the deliverers speak)
Daily Scrums

Each team member answers:
- What did you do yesterday?
- What are you doing today?
- What are your blocking issues?

No problem solving!

Leave after 15 minutes!
Daily Scrum Outcome

Records

- Sprint Backlog up to date
- Scrum Master updates the blocks list
Sprint Review Meeting

- Held the last day of the sprint
- Attended by team
- Team demos “done” user stories to stakeholders
  - Requests feedback
- Team holds retrospective
  - Updates the process for the next sprint
Demonstration

- Only DONE DONE working user stories.
- Ask for attendance from the following:
  - Executives
  - Internal users
  - Stakeholders
  - Customers
- Early iterations may be unsuitable for customers
- Add or Update Stories on the Release Backlog
Retrospective

Keep? Drop? Add?
What surprised us?
Other Retrospective Questions

- What was supposed to happen?
- What actually happened?
- Why were there differences?
Take Action

What process changes will we make for the next iteration?
The Team owns the learning from the retrospective.

They do not have to share it with the rest of the organization.
Scrum Exercise
Develop a Brochure in a 3-day Sprint

Complete Sprint Planning Meeting - 10min
- Select at least 5 Product Backlog Items
- Identify 2 to 3 Tasks per Item

Day 1
- 8 minute day

Day 2
- 2 minute Daily Scrum
- 8 minute day

Day 3
- 2 minute Daily Scrum Meeting
- 8 minute day

Demo & Reflection
Scrum Exercise
Retrospective
Agile Summary
Scrum

- Release Backlog
- Daily Scrum Meeting
- Product Backlog
- Sprint Backlog
- 24 Hours
- 2-4 Weeks
- Potentially Shippable Product Increment

Copyright © 2005, Mountain Goat Software
Scrum on a Page

Roles
- Stakeholders
- Product Owner
- Scrum Master
- Team

Artifacts
- Product Backlog
- Release Backlog
- Sprint Backlog
- Blocks List
- Information Radiator

Meetings
- Product Planning
- Release Planning
- Sprint Planning
- Daily Scrum
- Sprint Review Meetings

Agile is continuous learning and adaptive planning.

- M. Buckingham
A good agile project will build something that meets customers needs but may be different from original plans.

- Jim Collins
Make decisions together to avoid handoffs
Dev Team decides how – nothing technical in user stories
Cover all types of stakeholders
It’s all about learning
Pace yourselves
Don’t accumulate technical debt
Beware of chicken subversion
Getting Started Tips

- Expect the teams to over estimate in the first few sprints
- It will take about 5 sprints to develop a cadence and velocity
- Teams may take on too much after some time
- Watch out for anti-bodies
Things to Consider

- Management Support
- Strong and Experienced Leader(s)
- Picking the right project as a proof point
- Providing the right education, tooling and governance
- Ability to allow change to occur
- Keep it Simple
Scrum has been used by:

- Microsoft
- Yahoo
- Google
- Electronic Arts
- Lockheed Martin
- Philips
- Siemens
- Nokia
- IBM
- Capital One
- BBC
- Pitney Bowes

- Intuit
- Nielsen Media
- First American Real Estate
- BMC Software
- Ipswitch
- John Deere
- Lexis Nexis
- Sabre
- Salesforce.com
- Time Warner
- Turner Broadcasting
References

- http://scrumalliance.org/pages/what_is_scrum
- *Scrum and XP from the Trenches*, Henrik Kniberg
  http://www.infoq.com/minibooks/scrum-xp-from-the-trenches
- *The Elegant Solution*, Matthew May
- *Outside-in Software Development*, Carl Kessler and John Sweitzer
References

Agile Project Management with Scrum, Ken Schwaber (for scrum novices)

Agile Software Development with Scrum, Ken Schwaber and Mike Beedle (for experienced scrum types)
1. **WWW**: What Went Well

2. **WCBI**: What Could Be Improved

3. **NPS**: Net Promoter Score

   “Would you recommend this class to a peer?”

   10-9 Yes!
   8-7 Neutral
   0-6 No way!