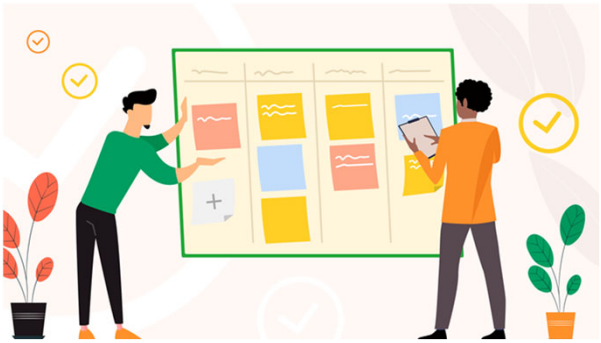


img

Planning + Estimating & Project Management

CSC 326

© NC State Software Engineering Faculty

1

"Days of coding can save you hours of planning."

- Dana Lasher

© NC State Software Engineering Faculty

2

Project Management

- The practice of coordinating a team of resources in the act of delivering on a set of customer requirements

- 3 Phases
  - **Planning**: scope, planning work, scheduling resources
  - **Execution**: performance of process
  - **Monitor**: evaluation and optimization of plan and execution

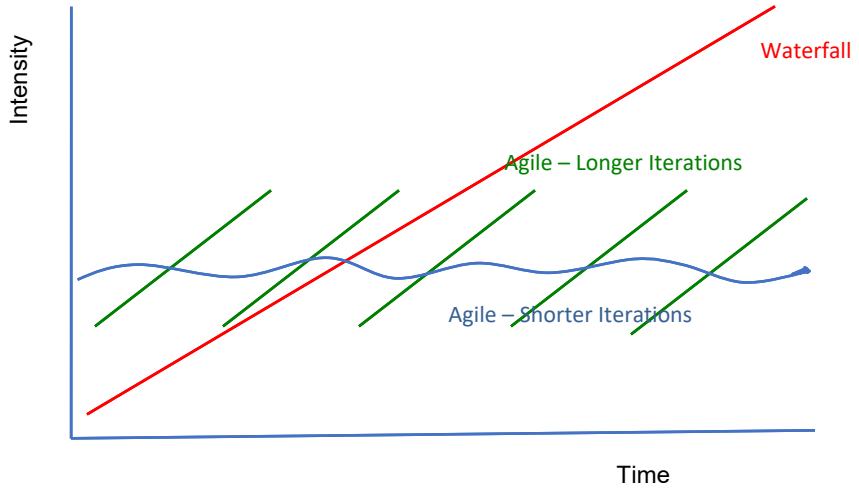


img

© NC State Software Engineering Faculty

3

Smoothing Team Intensity



Inspired by M. Cohn "Agile and the Seven Deadly Sins of Project Management" presentation

4



Planning

© NC State Software Engineering Faculty

5

Running Example

You work for a **dog walking company, WaltzingRover,** that has partnered with a **dog rescue** organization to provide free walks for foster dogs. WaltzingRover is a startup, operating within the I-540 beltline in Raleigh.

Currently, a WaltzingRover administrator **can assign dog walkers to dogs** based on the desired schedule of the dog's owner. Notifications are sent to dog walkers 30 minutes in advance of a walk reminding them of the upcoming walk appointment. Dog walkers send updates to the dog owners following the completion of a walk.

6

New Feature Requirements

To facilitate the partnership between WaltzingRover and the dog rescue, the WaltzingRover CEO has requested that if all of the following conditions apply, a foster dog walk should happen:

1. The dog walker has completed a walk with a paying client and the paying client is within 1 mile of a foster dog
2. At the completion of the paying client walk, the dog walker has 60 minutes or more before the next paying client.
3. The additional foster dog walk will not make the dog walker exceed 40 hours/week based on the paying client schedule
4. The foster dog has not been adopted before the foster walk

7

7

SWAG (Scientific, Wild Guess)

Assume you've been at WaltzingRover for 1 year and are familiar with the tech stack. With a couple people around you:

1. How long will it take for your group (those in the conversation right now) to complete this feature?
2. Why do you think that?

8

8

Task Planning

1. Task Identification and Breakdown
2. Estimates for task completion
   - Time (ideal hours)
   - Story point (unit-less relative measure)
3. Put together all estimation to create a schedule

Who/when:

- When a project is starting out, or after a major milestone
- Preferably involving the project group

© NC State Software Engineering Faculty

9

Example - CoffeeMaker

New customer requirement:

"*As a barista, I want to notify a customer's phone that their order is complete so I can complete my job expectations*."

Tasks for new requirements (examples only, not a complete list):

- System has a notification system that can send text messages
- Barista can mark a coffee order as "Done" in the system
- Each coffee order is associated with a customer's cell phone number

10

Task Identification

With your group, identify 3-5 **new tasks** required by WaltzingRover for the requested feature.

Exercise 11.01: Task Identification

If all of the following conditions apply, a foster dog walk should happen:

1. The dog walker has completed a walk with a paying client and the paying client is within 1 mile of a foster dog
2. At the completion of the paying client walk, the dog walker has 60 minutes or more before the next paying client.
3. The additional foster dog walk will not make the dog walker exceed 40 hours/week based on the paying client schedule
4. The foster dog has not been adopted before the foster walk

11

11

Estimating

- **Estimating** – estimating the [**resources**, **time**, **size**] required to develop a [**feature** or **requirement**]
- **Planning** – putting the estimates together to formulate a project plan and schedule

© NC State Software Engineering Faculty

12

12

Benefits of Task Planning

- Definition of Project Scope
  - Identify expectations of instructors
- Illumination of Project Deliverables
  - Evolve culture of software development beyond “writing code”
- Accountability for teams
  - Reduces conflict by identifying tasks
  - Increases leadership (including technical leadership and ownership)

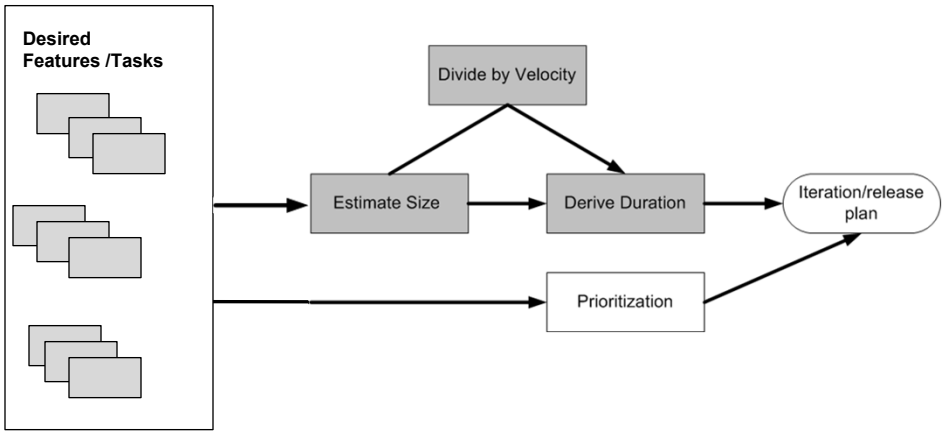
© NC State Software Engineering Faculty

Task Planning

- A task is a specific “to do” to complete a story, feature, requirement, artifact…
- Owners assigned
- Estimated in **ideal time** or **story points**
  - By team or individuals
- Estimates typically entered into a project management system & tracked daily (requiring a few minutes/day to provide status)
  - You’re doing a little of this for the OBP!
  - You’ll do more for the Team Project!

© NC State Software Engineering Faculty

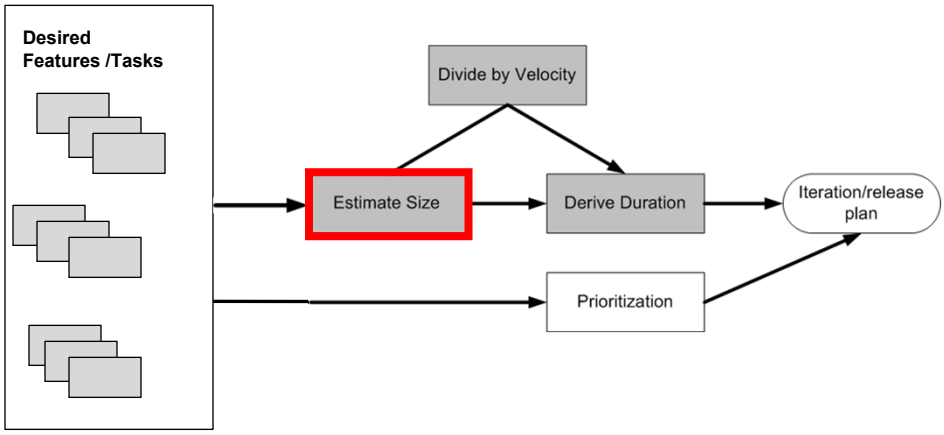
Coming up with the plan



© NC State Software Engineering Faculty

15

Coming up with the plan



© NC State Software Engineering Faculty

16

Estimating size: concepts



- Ideal time: the amount of time "something" takes when stripped of all peripheral activities
  - Example: American football game = 60 minutes
- Elapsed time: the amount of time that passes on the clock to do "something"
  - Example: American football game = 3 hours



- Story point: unit of measure for expressing the **overall size of a user story, feature,** or other piece of work. The raw value of a story point is unimportant. ***What matters are the relative values.***
  - Related to how hard it is and how much of it there is *compared to other requirements*
  - NOT related to amount of time or # of people
  - Unitless, but numerically-meaningful
- Velocity: measure of a team's rate of progress

© NC State Software Engineering Faculty

Estimating ideal time

- Ideal time vs. elapsed time in software development
  - Supporting current release
  - Sick time
  - Meetings
  - Demonstrations
  - Personal issues
  - Phone calls
  - . . . .
- When estimating ideal time, assume:
  - The feature being estimated is the only thing you'll work on
  - Everything you need will be on hand when you start
  - There will be no interruptions

© NC State Software Engineering Faculty

Ideal Time

For your team of WaltzingRover employees, how long will EACH of the following tasks take in ideal days?

1. WaltzingRover has access to locations of current foster dogs
2. WaltzingRover can calculate distances from the paying client house to the foster dog house.
3. WaltzingRover contains the hours worked for each of the dog walkers, and the planned hours for a given week.
4. An hour before the foster dog walk occurs, WaltzingRover needs to ping the Rescue API and ensure the dog has not been adopted.

Exercise 11.02: Ideal Days

19

19

Estimating story points

- Choose a medium-size story and assign it a "5"
- Estimate other stories relative to that
  - Twice as big
  - Half as big
  - Almost but not quite as big
  - A little bit bigger
- Only values:
  - 0, 1, 2, 3, 5, 8, 13, 20, 40, 100

Near term iteration "stories"	A few iterations away "epic"

© NC State Software Engineering Faculty

20

20

Story Points

For your team of WaltzingRover employees, how long will EACH of the following tasks take in Story Points?

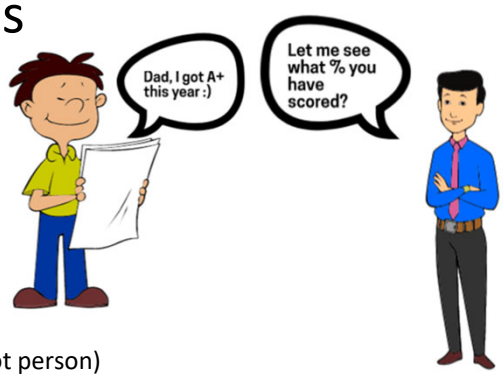
1. WaltzingRover has access to locations of current foster dogs
2. WaltzingRover can calculate distances from the paying client house to the foster dog house.
3. WaltzingRover contains the hours worked for each of the dog walkers, and the planned hours for a given week.
4. An hour before the foster dog walk occurs, WaltzingRover needs to ping the Rescue API and ensure the dog has not been adopted.

Exercise 11.03: Story Points

21

21

Ideal days vs. Story points



- Favoring ideal days:
  - Easier to explain outside of team
  - Estimation is typically faster at first
- Favoring story points:
  - Help drive cross-functional behavior
  - Do not decay (change based on experience)
  - Are a pure measure of size (focus on feature, not person)
  - Estimation is typically faster in the long run
  - My ideal days are not your ideal days (focus on person and their speed ☹)

- See http://agilesoftwaredevelopment.com/2006/12/measures-of-size for other size measures

© NC State Software Engineering Faculty

22

22

Deriving an estimate for a requirement

- Expert opinion
  - Rely on gut feel based on (extensive) experience
  - Disadvantage for agile: need to consider all aspects of developing the user story, so one expert will likely not be enough
- Analogy
  - Relative to (several) other user stories
  - Triangulation: little bigger than that "3" and a little smaller than that "8"
- Disaggregation / Task Decomposition
  - Break up into smaller, easier-to-estimate pieces/tasks.
  - Need to make sure you don't miss any tasks.
  - Sanity check: does the sum of all the parts make sense?
- **Planning poker**
  - Combines expert opinion, analogy, disaggregation

© NC State Software Engineering Faculty

Playing Planning Poker



- Include all players on the development team (but less than 10 people overall):
  - Programmers
  - Testers
  - Database engineers
  - Requirements analysts
  - User interaction designers . . .
- Moderator (usually the product owner or product manager) reads the description and answers any questions
- Each estimator privately selects a card with their estimate
- All cards simultaneously turned over
- Re-estimate
- Repeat until converge (typically ≤ 3votes)

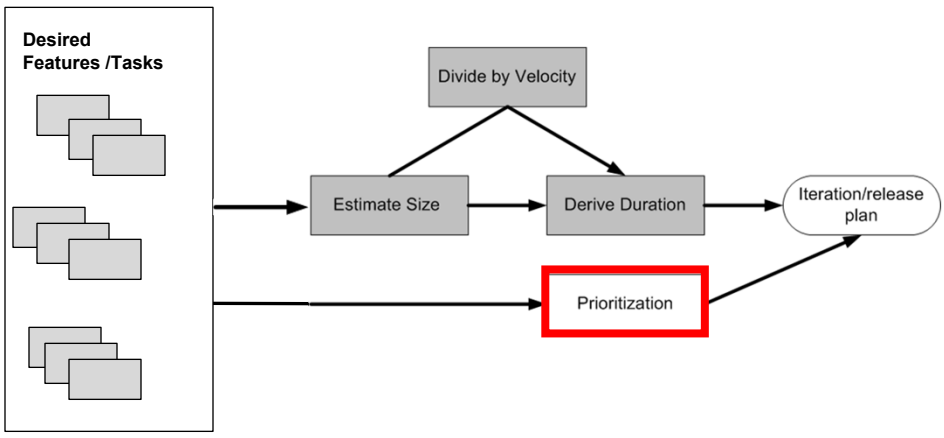
© NC State Software Engineering Faculty

(Subjective) Value of Planning Poker: It's Not only About the Number!

- Explicit result (<20%):
  - Effort Estimate
- Side effects/implicit results (80%+):
  - Greater understanding of requirement
  - Expectation setting
  - Implementation hints
  - High level design/architecture discussion
  - Leveling of value versus cost: product owner uses their precious velocity budget
  - Ownership of estimate

© NC State Software Engineering Faculty

Coming up with the plan



© NC State Software Engineering Faculty

Prioritization

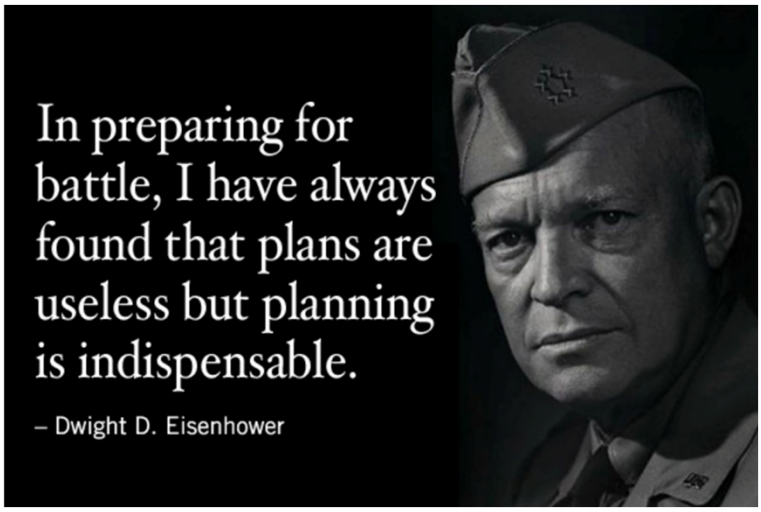
- Driven by customer, in conjunction with developer
- Choose features to fill up velocity of iteration, based on:
  - Desirability of the feature to **a broad base of customers**
  - Desirability of a feature to a **small number of important customers**
  - The cohesiveness of the story in relation to other stories. Example:
    - "Zoom in" a high priority feature
    - "Zoom out" not a high priority feature
      - But it becomes one relative to "Zoom in"
- Priority
  - High – critical, must have feature
  - Medium – would like to have
  - Low – stretch goal

© NC State Software Engineering Faculty

Prioritization

How would you prioritize these features? Talk with your group and create an ordering.

1. WaltzingRover has access to locations of current foster dogs
2. WaltzingRover can calculate distances from the paying client house to the foster dog house.
3. WaltzingRover contains the hours worked for each of the dog walkers, and the planned hours for a given week.
4. An hour before the foster dog walk occurs, WaltzingRover needs to ping the Rescue API and ensure the dog has not been adopted.



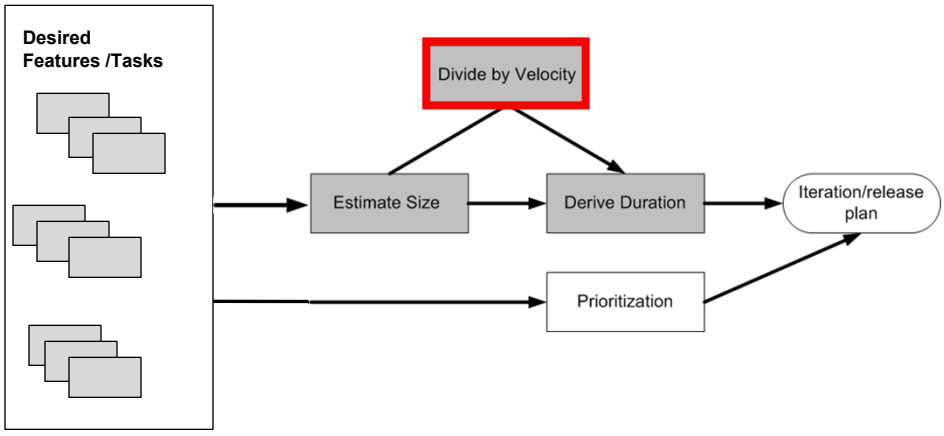
img

Execution

© NC State Software Engineering Faculty

29

Coming up with the plan

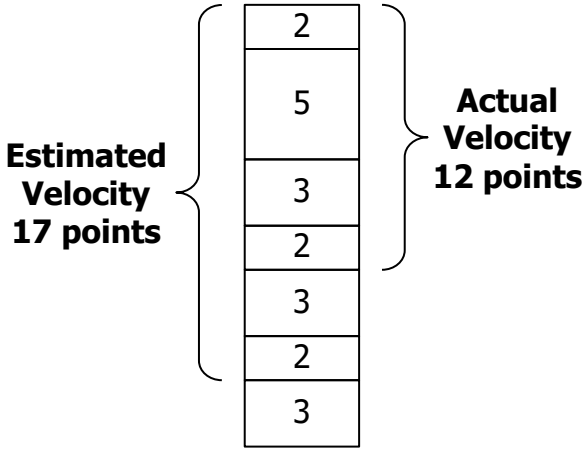


© NC State Software Engineering Faculty

30

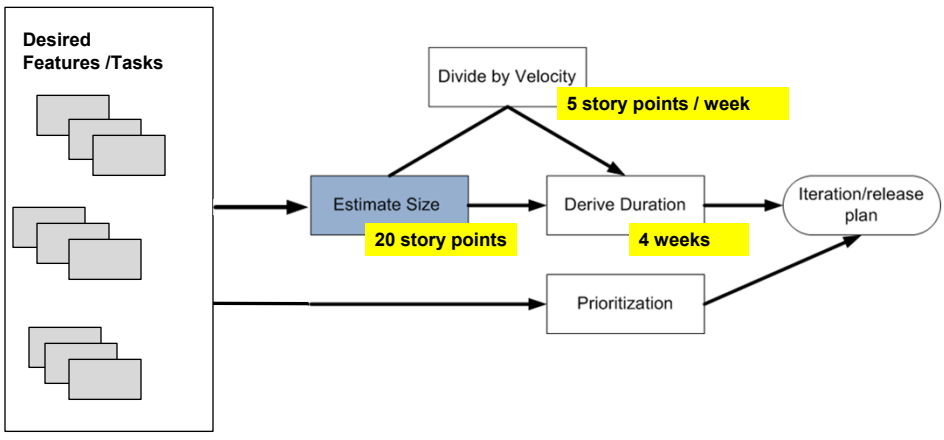
Velocity: Points Completed in an Iteration

- Velocity is a measure of a **team's** rate of progress.
- Velocity is calculated by summing the number of story points assigned to each user story that the team completed during the iteration (it's a retrospective number).
- We assume that the team will produce in future iterations at the rate of their past average velocity.
  - "Yesterday's weather" https://www.scruminc.com/yesterdays-weather/



© NC State Software Engineering Faculty

Coming up with the plan



© NC State Software Engineering Faculty

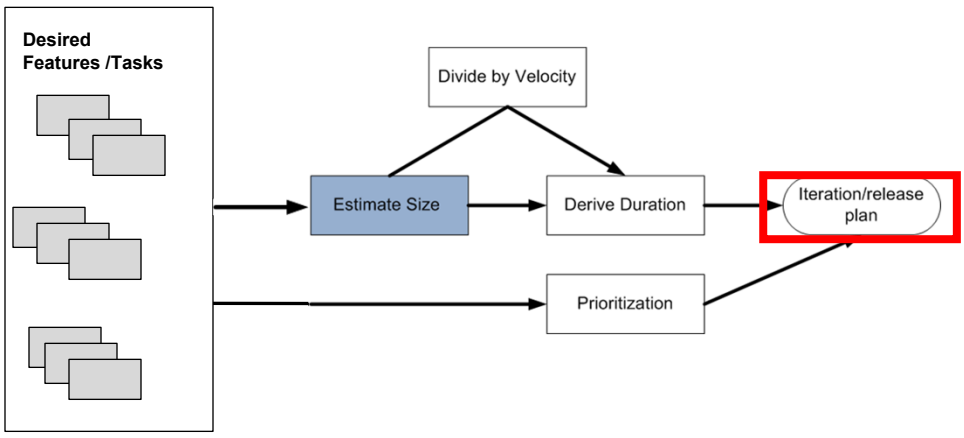
Monitor

evaluation and optimization of plan and execution

33

33

Coming up with the plan



© NC State Software Engineering Faculty

34

34

Iteration Length

- Generally 2-4 weeks, sometimes 1 week
  - Industry trend shorter (2-3 weeks)
- Why shorter?
  - Easier to defer priority changes to next iteration
  - Easier to estimate (gut feel … can we do this in the next two weeks or less?)
- Why longer?
  - May be difficult to break features into smaller pieces
  - Iteration cadence meetings happen less frequently

35

35

Release Planning

- “If you don’t know where you are going, it's easy to iteratively not get there.”
  - David Hussman, DevJam



www.devjam.com

36

36

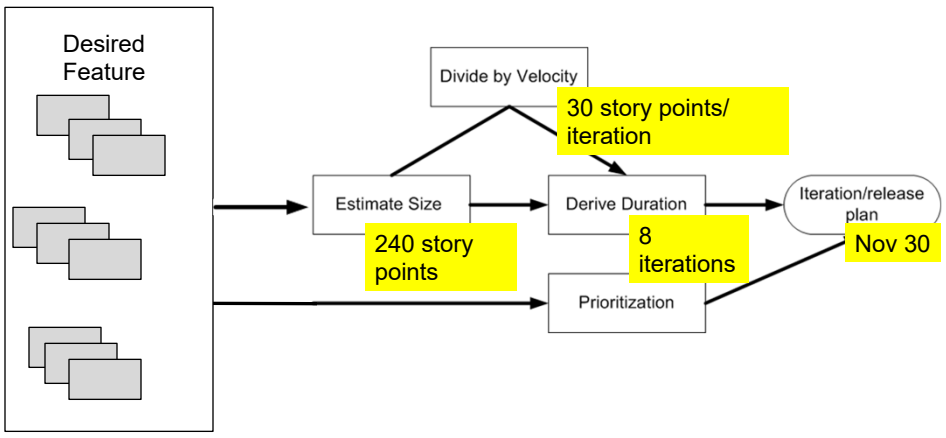
Release Planning Input

- Input (more on each of these):
  - High level epics/stories/requirements, grouped into themes if possible
  - Project Goal:
    - Desired release date (date-driven plan)
    - Desired functionality (feature-driven plan)
  - Planned velocity
  - Iteration length

37

37

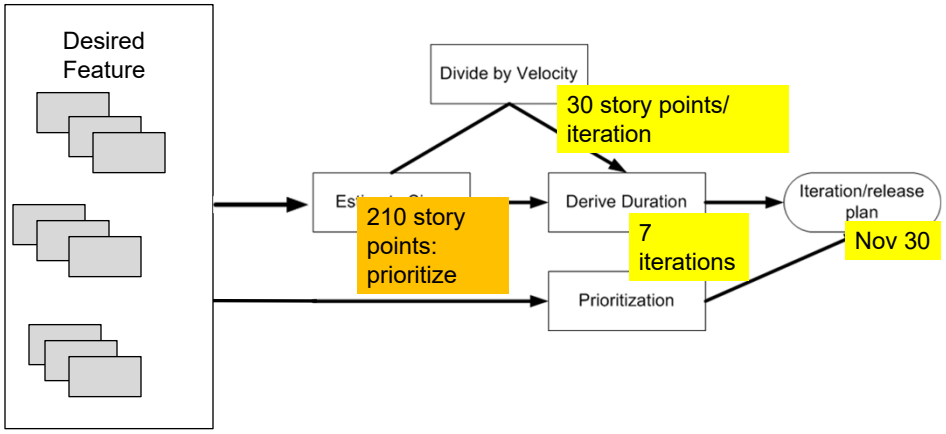
Feature-driven plan



38

38

Date-driven plan



39

39

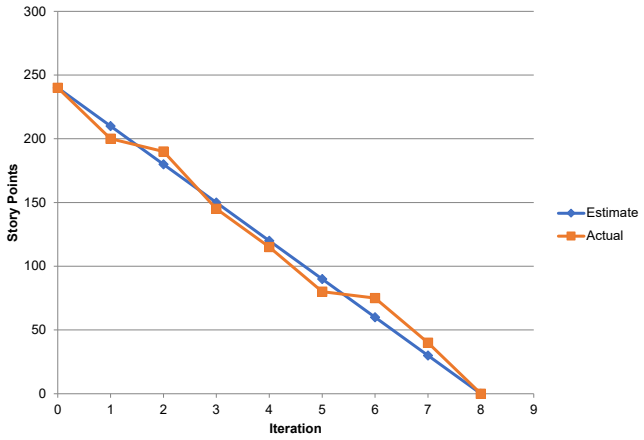
Release Planning Processing

- Primarily done by key leaders
- Planning poker for resources for epics/stories
- Determination if end date and/or scope is feasible and prudently commit-able
  - Adjust and prioritize accordingly
  - Put accepted epics/stories in release backlog
- Prioritization of themes to assign epics/stories to iteration plan

40

40

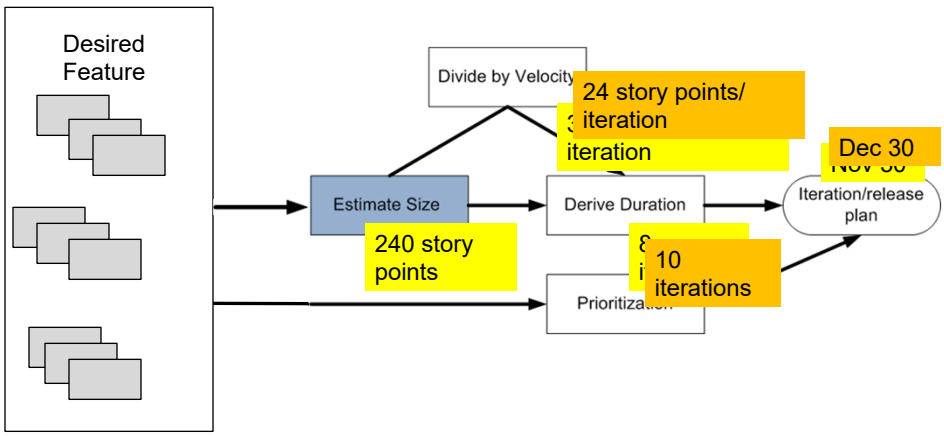
Release burndown



41

41

Not working as fast as planned?



42

42

Iterations + Done Criteria

Iteration 0

- Team makeup; Organizational considerations
- Non-functional requirement considerations
- Requirements envisioning
- Architecture envisioning
- User interface template development, “look and feel” development
- Preliminary database structure
- Build and test environments built
- (Anonymized) test database established
- Continuous integration/deployment system(s) in order
- Establish team rules, eg. mini-feature freeze deadlines for each iteration
- Workroom / collaborative spaces set up
- Other tool support
- Establish “done” criteria necessary to earn story points (next slide)

Iteration Rhythm

[Iteration N]

- [Prior] Product manager, development manager, UI discuss more than enough stories in backlog for Iteration N
- Demonstrate Iteration N-1 functionality to customer (acceptance tests plus more)
- Perform retrospective on Iteration N-1
- Determine velocity for next iteration ("yesterday's weather", vacations?, holidays?, classes?, unfinished stories)
- Estimate Iteration N stories using Planning Poker
- Choose highest priority stories to fill velocity
- [After] Have task planning meeting (if not done in conjunction with Planning Poker)
- [After] Test + customer/rep + developer create acceptance tests

45

45

Typical "Done" Criteria

- Acceptance tests pass and can be demonstrated in iteration planning meeting
- 80% automated unit test statement coverage; all pass
- All system/acceptance tests have been run by testers
- No high priority defects are open
- Static analysis tools run & appropriate faults fixed
- Design and/or code review completed
- Documentation is "complete"
- Acceptance (and other) tests are automated

- Focus on building quality in throughout lifecycle
  … and don't build **technical debt**
  … be in a "potentially shippable" state at the end of each iteration



img

46

46

Other kinds of debt…

Think - Pair - Share: have you ever intentionally (or unintentionally) introduced **technical debt** into software?

"I'll put these new shoes on my credit card and pay it off when I get my next paycheck" – financial debt

"My friends are coming over but my house is a mess. I'll throw everything in the closer and deal with it later." – organization debt

"There's a leak in my ceiling but I'm late to class. I'll put an obnoxious bucket in the middle of the room to avoid more damage now, and then I'll investigate the root cause after class" – house repair debt

47

47

Our "Done" Criteria

- Appropriate requirements describing the element.
- Design describing change to the system and any updates to a planned design.
- Working demo of functionality that meets requirements and management approval.
- Front-end should consider accessibility in regards to color/shapes, tab order, and tool tips to support screen readers.
- Passing unit tests that cover 60% of statements.
- Passing system tests.
- Passing acceptance tests as outlined in system test plan (and that preferably automated).

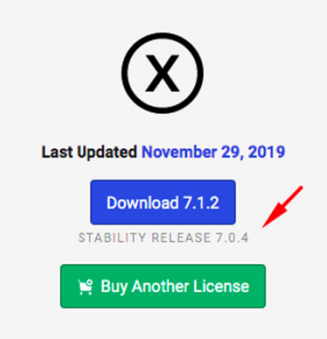
© NC State Software Engineering Faculty

48

48

Stabilization Iteration

- May be good for longer releases (≥ 6 months long)
- Perhaps every 3 months
- Generally, two weeks regardless of other iteration length
- Essentially an Iteration 0 for next iteration grouping
- More testing
  - Performance testing
  - Regression testing
  - Integration testing
- Bring down defect count
- **Motivation**: reduce burn-out caused by iterations ad infinitum
- **But….** not a generally accepted practice in agile community
- No story points are earned during the stabilization iteration



img