Estimating and planning

- Estimating – estimating the [resources, time, size] required to develop a [user story, feature, or requirement]

- Planning – putting the estimates together to formulate a project plan and schedule
**Estimating size: concepts**

- **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
  - NOT related to amount of time or # of people
  - Unitless, but numerically-meaningful
- **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
- **Velocity**: measure of a team’s rate of progress

---

**Coming up with the plan**

1. **Desired Features**
2. **Estimate Size**: 30 story points
3. **Divide by Velocity**: 5 story points/two-week iteration
4. **Derive Duration**: 6 iterations
5. **Iteration/release plan**: 30 October
6. **Prioritization**
Estimating “dog points”

- Estimate each of the dogs below in dog points, assigning each dog a minimum of 1 dog point and a maximum of 10 dog points
- A dog point represents the height of a dog at the shoulder
  - Labrador retriever
  - Terrier
  - Great Dane
  - Poodle
  - Dachshund
  - German shepherd
  - St. Bernard
  - Bulldog

What if?

- Estimate each of the dogs below in dog points, assigning each dog a minimum of 1 dog point and a maximum of 100 dog points
- A dog point represents the height of a dog at the shoulder
  - Labrador retriever
  - Terrier
  - Great Dane
  - Poodle
  - Dachshund
  - German shepherd
  - St. Bernard
  - Bulldog
  - Harder or easier?
  - More or less accurate?
  - More or less time consuming?
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

Estimating ideal days

- Ideal days vs. elapsed time in software development
  - Supporting current release
  - Sick time
  - Meetings
  - Demonstrations
  - Personal issues
  - Phone calls
  - ....
- When estimating ideal days, assume:
  - The story 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
Deriving an estimate for a user story

- **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**
  - 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

---

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 analyst) reads the description and answers any questions
- Each estimator privately selects a card with their estimate
- All cards simultaneously turned over
- Discussion ensures
- Re-estimate
- Repeat until converge
Velocity

- 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 operation.
- We assume that the team will produce in future iterations at the rate of their past average velocity.

— “Yesterday’s weather”
Not working as fast as planned?

Velocity corrects estimation errors

- Since all user stories are estimated relative to each other . . .
- It’s the velocity that should change, not each story point estimate for future releases
**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 or users
  - Desirability of a feature to a small number of important customers or users
  - 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”

- High Priority – “Give us these stories to provide a minimal working system.”
- Medium Priority – “We need these stories to complete this system.”
- Low Priority – “Bells and whistles? Which stories can come later?”
Coming up with the plan

Burn down charts
- Vertical axis shows number of story points or hours remaining in the project
- Iterations [or days] are shown across the horizontal line
- Shows the amount of work remaining at the start of each iteration
- Visual indicator of how quickly a team is moving toward its goal