Capacity Planning

"How to Retire Early by Spending 2 Hours Forecasting the Future"

Agenda

- Overview
  - Planning Release
  - Capacity Planning
- Availability – The Foundation
  - Assumptions
  - Inputs
- Adjustments
- Planning Capacity
- Other Aspects

Reality

- OK, we won't be teaching you about how to retire early. We will be talking about predicting the future
- If all of my efforts to retire early by predicting the future have failed, why predict the future?
  - "Plans are useless, planning is indispensable"
- Just because our plans will be inaccurate to some extent, that does not mean we should abandon planning
- We should be able to set some expectations on cost and timeframes while recognizing that reality will be different

Overview - Questions to Answer

- When are you going to be done?
  - Reliably predict end dates
    - Backlog - how much work do I have?
    - Velocity - how quickly can I get the work done?
    - Combination tells me when I will be done
- How much will this cost me?
- What will I get?
- Is my team getting better over time?
  - Performance
  - Quality
- How effective is my team?
  - Story time, defect time and "overhead" time
- Do we have special efforts around elevates/deployments?

Overview - Primary Focus

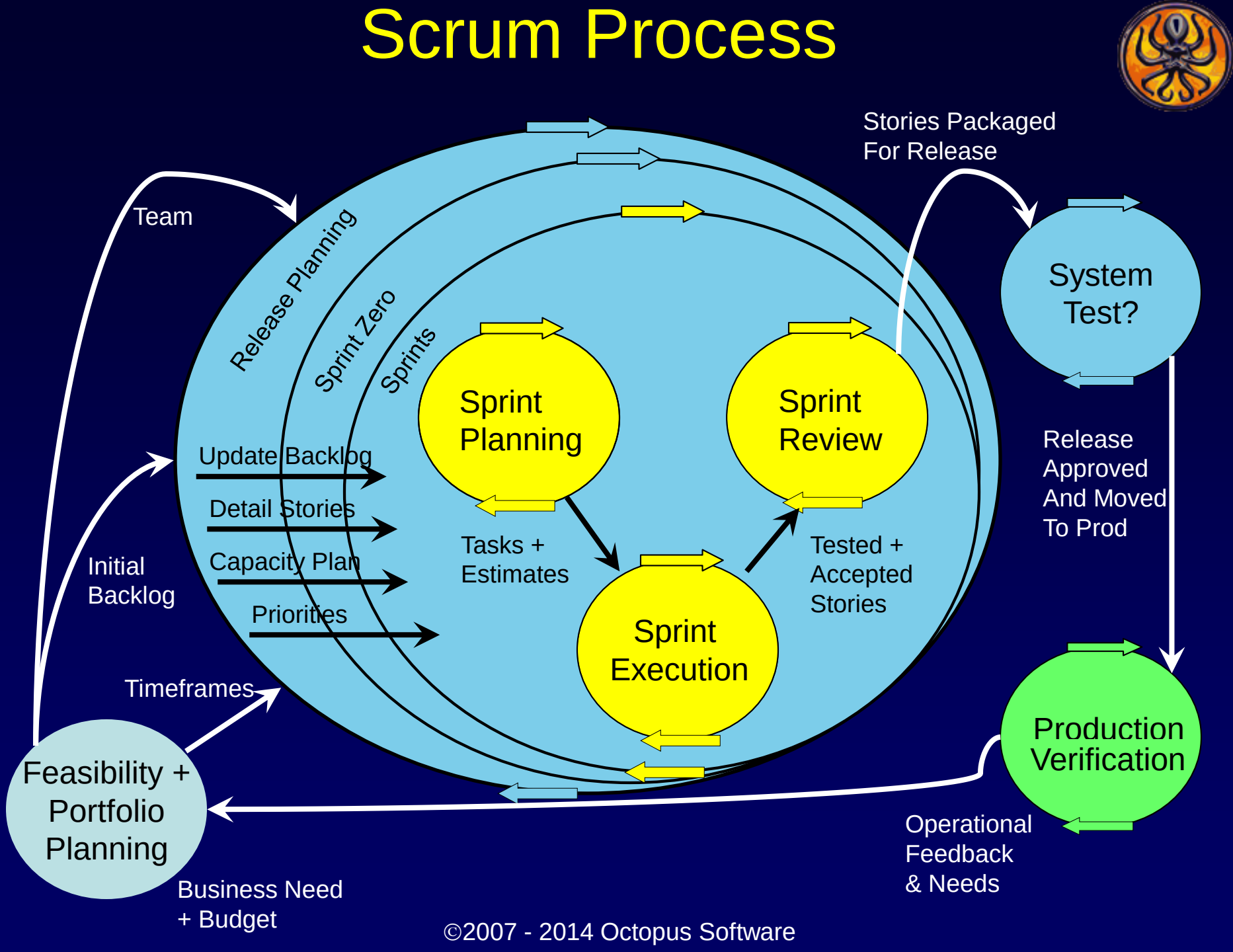
Key Question: “When can I predict my team to deliver with high quality and reliability?”

Two Part Answer:

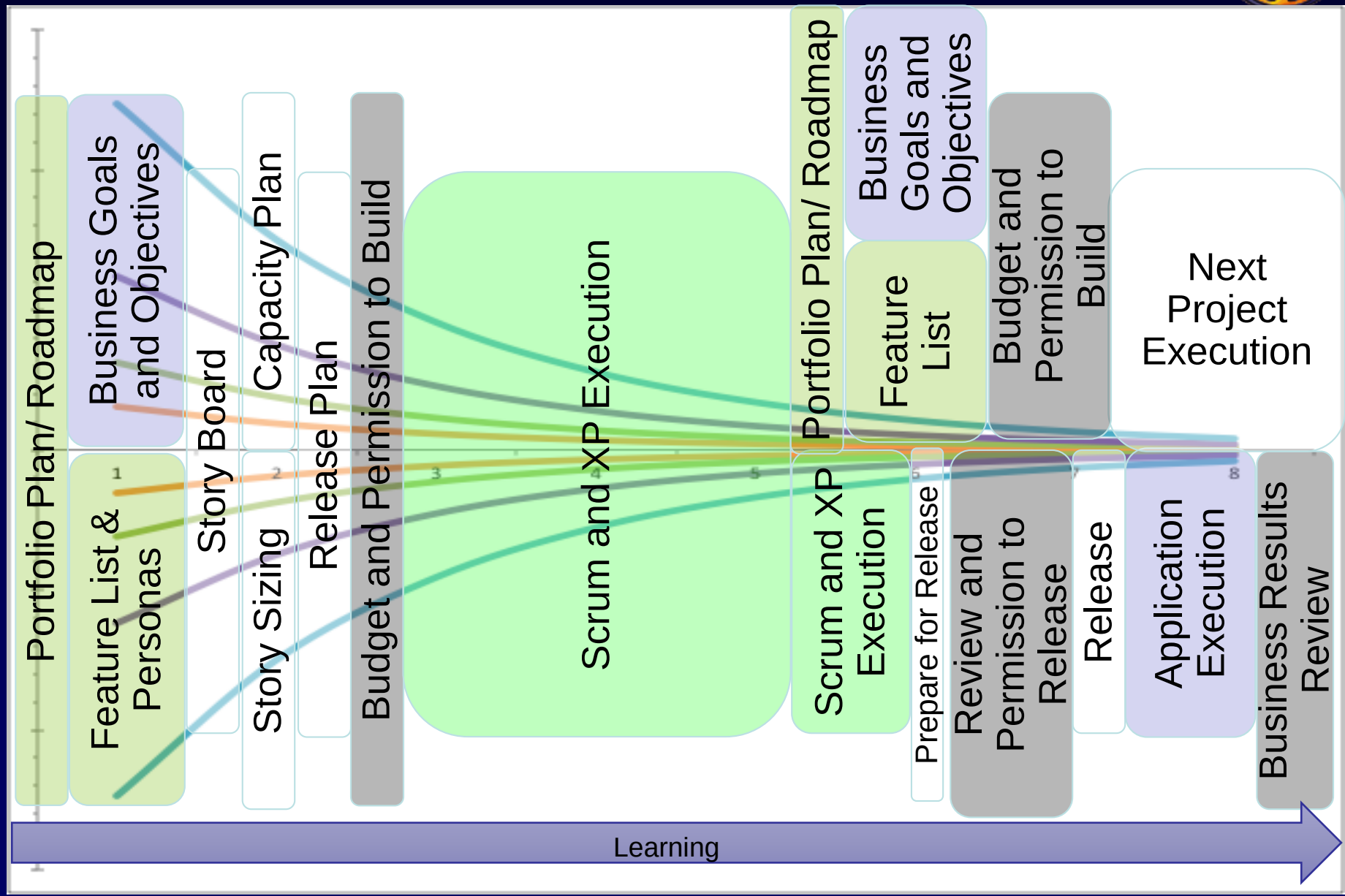
– Capacity - How much can my team create and delivery with high quality?
  - Availability - how much time does my team have and how efficient are they?
  - Defects - how do I address and manage this effort?
  - Ongoing velocity - am I getting better or worse?
  - Team composition - expertise, experience and “heart”
  - Predicted with a Capacity Plan, predicted with Story Points and evidence of previous performance

– Release Backlog - How much work do I have?
  - Sized by estimating Stories and Epics with Story Points

Scrum Process



Overview - Cadence + Risk



Overview - What Is Capacity?

- To understand capacity, we want to think of our team as a factory to build intellectual property
  - Not widget builders
  - Not typists
  - But inventors, knowledge workers
- So how do we measure the invention of software? Lets take a look at a software factory

High Level Software Factory

Project Backlog

Velocity

One Cycle

C, size 1 | J, size 2 | N, size 1 ADDT | V, size 2

D, size 2 | G, size 4 | K, size 2 | O, size 1 ADDT | R, size 1 | AA, size 1

A, size 2 | E, size 2 | H, size 2 | L, size 2 | P, size 2 ADDT | S, size 1 | B2, size 4 | CC, size 2 | AB, size 2

B, size 4 | F, size 1 | I, size 2 | M, size 2 | Q, size 4 ADDT | U, size 2 | BB, size 1 | DD, size 2 | AC, size 2

Capacity

Week 3 | 5 | 7 | 9 | 11 | 13 | 15 | 17 | 19

sprints

6	6	4	8	9	4	8	4	4
Velocity -	Actual			Planned	(Thanksgiving)		Last 2 sprints	

Original Backlog - 53 points

Now

Release Preparation

Evolving Design

System + Performance Test

End date = Backlog divided by Planned Capacity (Velocity)

Overview - What Impacts Capacity?

- Team composition and structure
- Expertise and experience
- Tools and technology
- Participation of our users/ business partners
- Definition of done
- Our experience with development and delivery tools
- Our experience with the business area

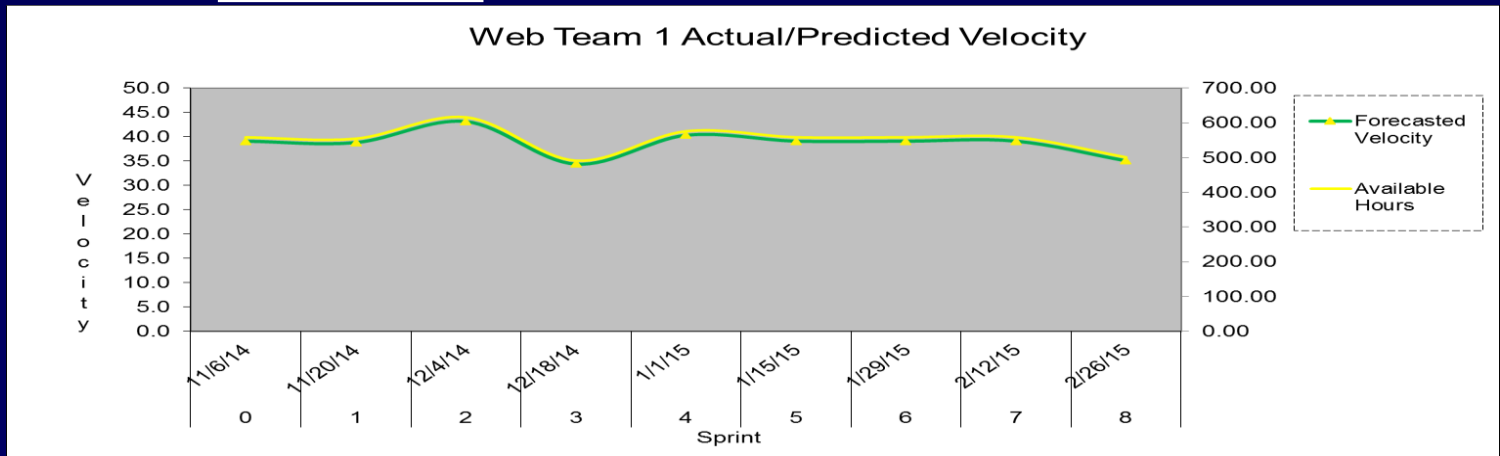
Overview – Team Time

Co-located and Complete Team

Each Person Contributes Effort Over Time

As available time goes up, so does the potential to deliver.
Our goal is to predict velocity based on that information.

Minus [Holidays, Vacations, Training, Other Projects, Other Absences] Minus [Defect Time, Support] Gives [Available Time]



Lets Get Started!

- We can plan capacity for 1 to 12 months in < 2 hours
- Preparation:
  - Download the spreadsheet
  - Open the spreadsheet
  - Put in your details and assumption
  - See your capacity plan in story points
  - We will start with some assumptions

Setup – Settings Tab

Define the basic sprint information – open the Settings tab:

- Starting number and date
- Sprint length in days
- Hours per Sprint
- Number of hours per sprint/day

Sprint Holiday time:

- Confirm sprint start and end dates
- Holidays for the year

You should notice a pattern where the green cells are cells you enter data.

Setup – Team Tab

Define the team:

- Person's name, role, start and end date
- % commitment to the team
- The start/end dates are computed
- Hours available per person are computed

Define vacation/ramp up time:

- By person's name, date and days off with comment
- Hours off and sprint are computed (non-green colored fields)

Foundation – Availability Tab

- Define the team allocation:
- Ensure the Team Allocation (column C) matches the summary by team (F1)
- Review the Holiday and Vacation values
- The available time per sprint and an FTE total is computed

Setup - WebTeam1 Tab

Name for the team:

- Change the values in A1, the tab and the graphs to match your team name

Hide rows to control x-axis on charts:

- Notice that the model can handle many sprints

Ensure Planning Velocity Ratio is appropriate:

- Setting this value requires some judgment from the Scrum Master
- Note that the Historical Values should start empty

Original Velocity Estimate

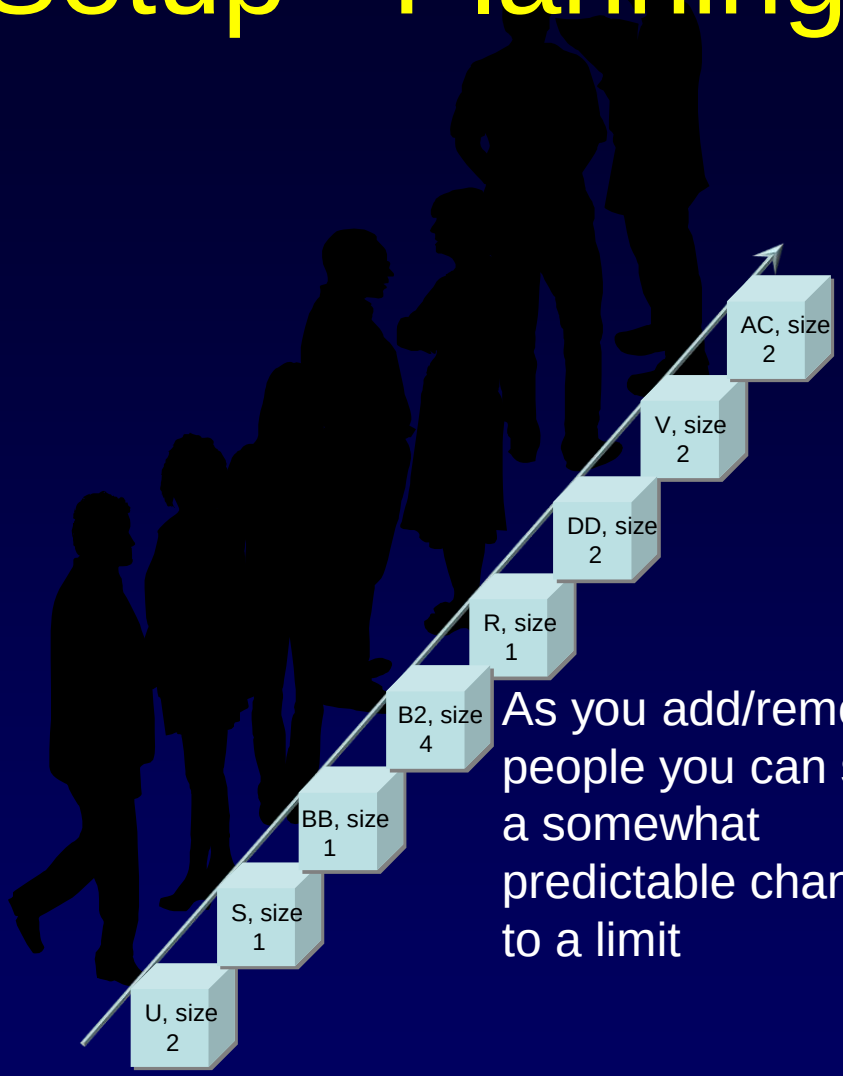
- Questions:
  - Prior experience
    - As a team
    - With specific tech
    - In business area
  - Iteration length
  - Point factor
- Pointing Model - 1, 2, 4 and 8, where 4 is two weeks of work, then factor in iteration length
- Initial Estimate Calculation

  3 Developers x 4 point/iteration = max 12 team points, so reduce 40% as target for first iteration ~ 7 points

Calculate Hours to Points Factor

- Now that we have an estimated velocity (7 points), we can create a conversion factor
- 3 developers @ 40 hours / week and a two week iteration = 240 total hours available
- 240 hours / x hours/point = 7 points
- Conversion factor = 34.28 hours/point
- If we loose one developer for 5 days of vacation the next iteration, we will be down to 200 hours
- For this next iteration, 200 hrs/ 34.28 = 5.8 points. Round down for RPM

Setup - Planning Velocity Ratio



As you add/remove people you can see a somewhat predictable change, to a limit

If you have good balance in your team and with your eco-system, you can avoid artificial throughput constraints

Setup – Velocity Stealers

Velocity “stealers” need to be set:

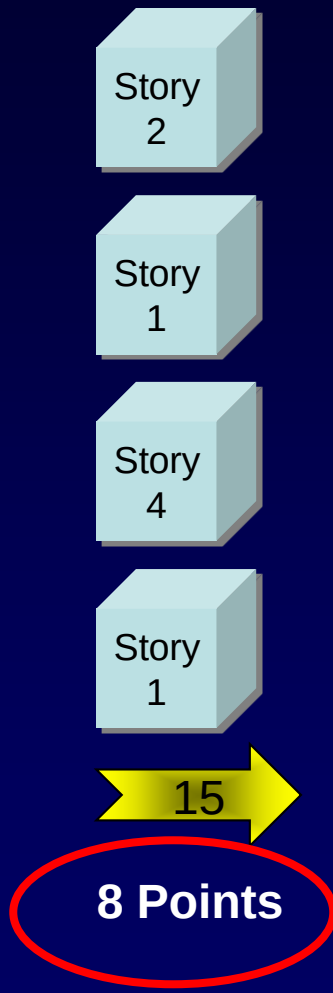
- Note Defect Hours to reflect our best understanding of how many hours per iteration we need for defects (column D)
- Note Support/Other Hours to reflect our best understanding of how many hours per sprint we need (column E)

Velocity Forecast

- The model has computed a Velocity forecast based on the information we have put in. This forecast is in Column P
- Recall that this is a plan of capacity and this does not include any risk-based analysis on other potential causes of velocity change
- This is a primary input into release planning. We need to review this with the team to see if this forecast makes sense.
- An S-Curve adjustment is possible for teams that are just starting up. This can adjust the specific sprint forecast in column P

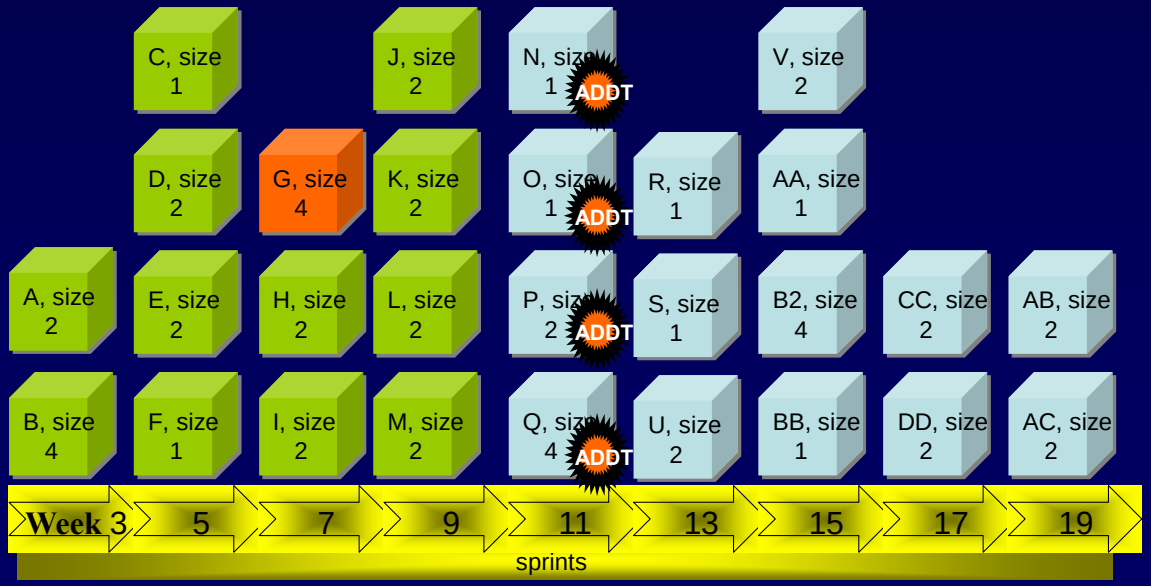
Capacity / Planned Velocity

- Capacity or velocity is a prediction of how much throughput my team can achieve
- Simply, this is based on "yesterday's weather" - what they did last iteration is what they will do now.
- The team may request to adjust it based on what they know
- The iteration manager may adjust it, with the team, based on what they know



Use of Forecast

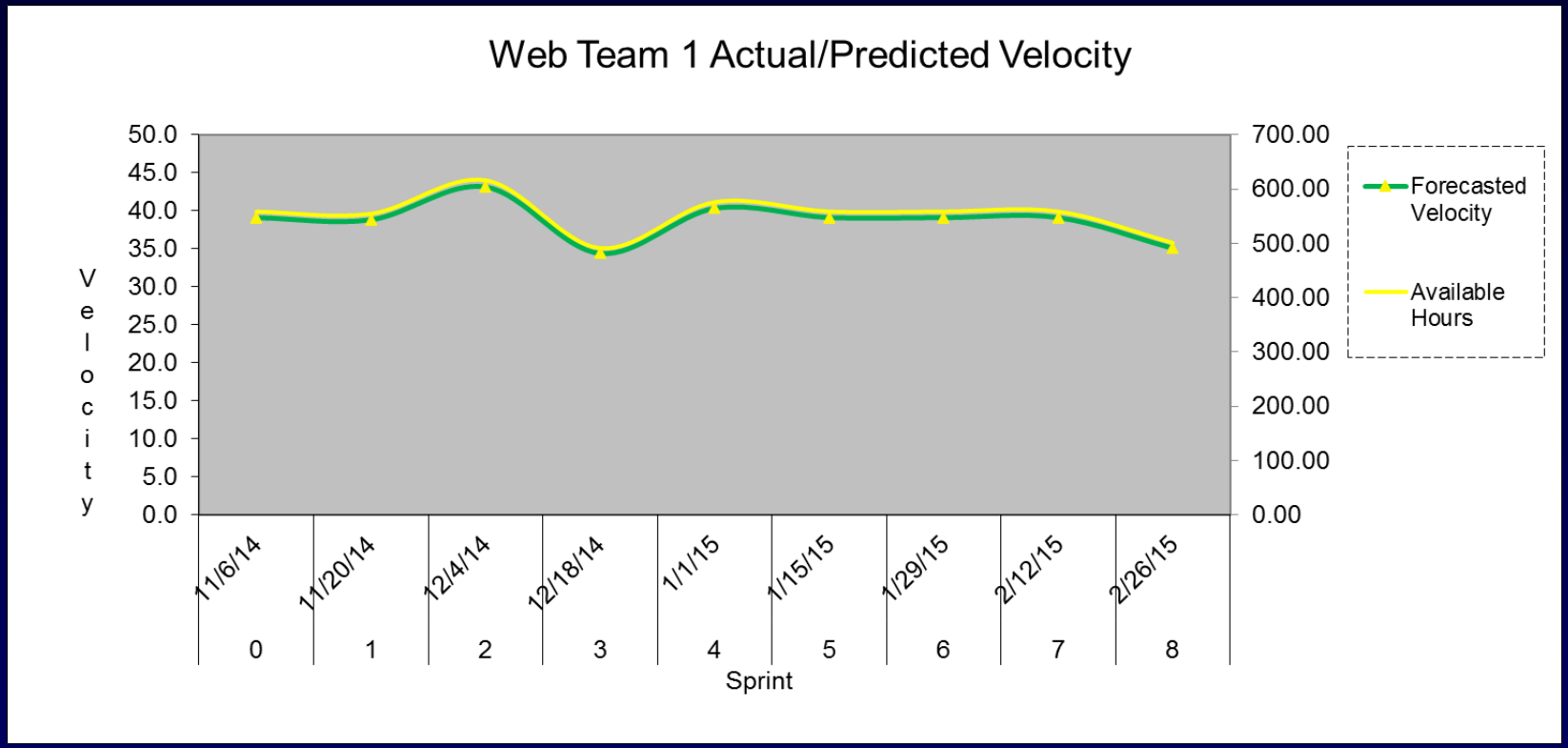
- Based on a desired end date, we can forecast what the total number of story points are possible
- We can have very interesting discussions on what could be done in each sprint to set detailed expectations and cross-team coordination



Adjusted or Requested Velocity

- Management, or the team, may request a level of throughput higher than what the model says. For whatever reason of variance, this value can be recorded in the Scheduled Story Points, column H
- The Total Points requested for this timeframe is in cell H41. Our planned and requested points are in sync in this example

Picture View of Plan



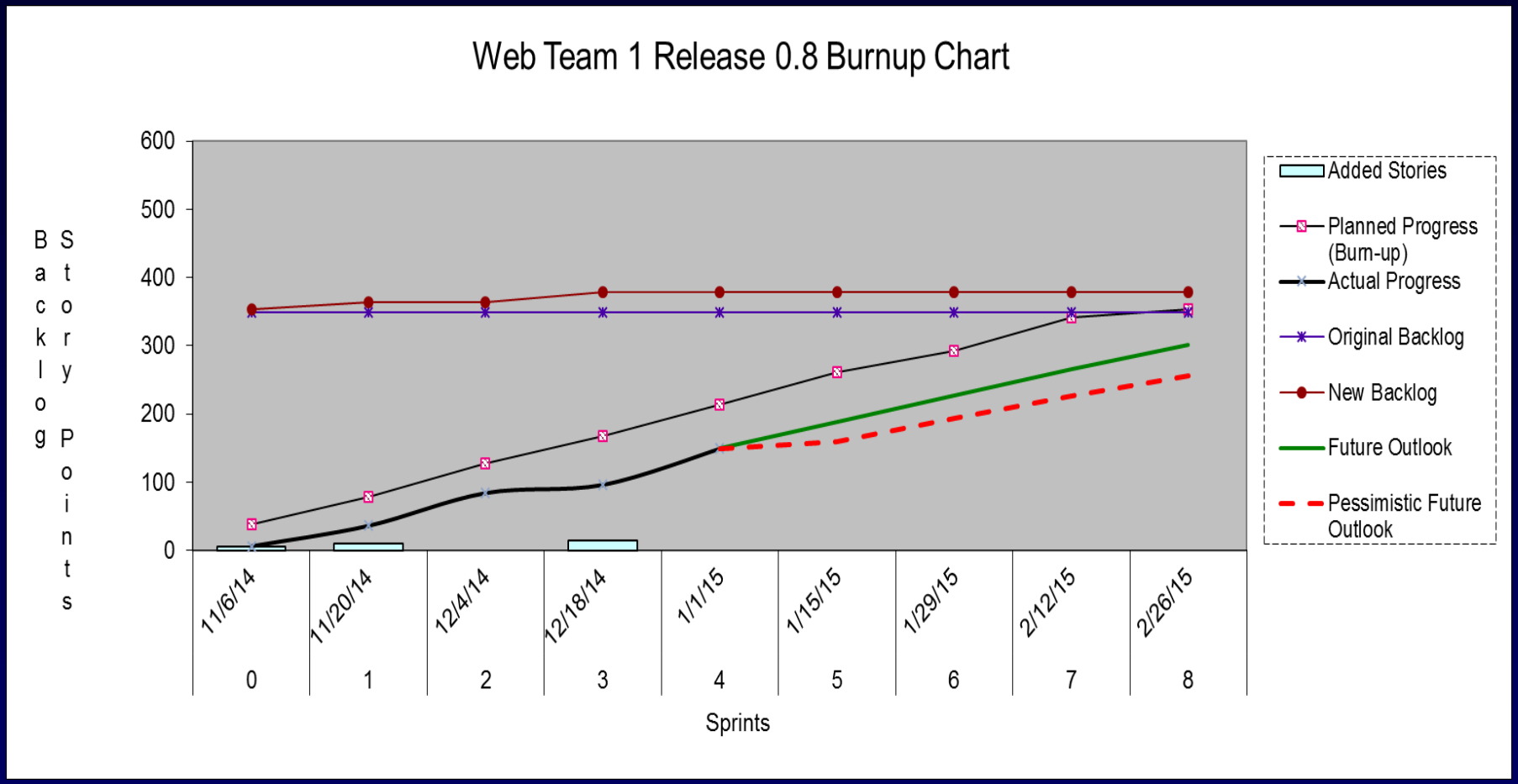
As hours available change, then the planned output of the team changes

Tracking Progress To Plan

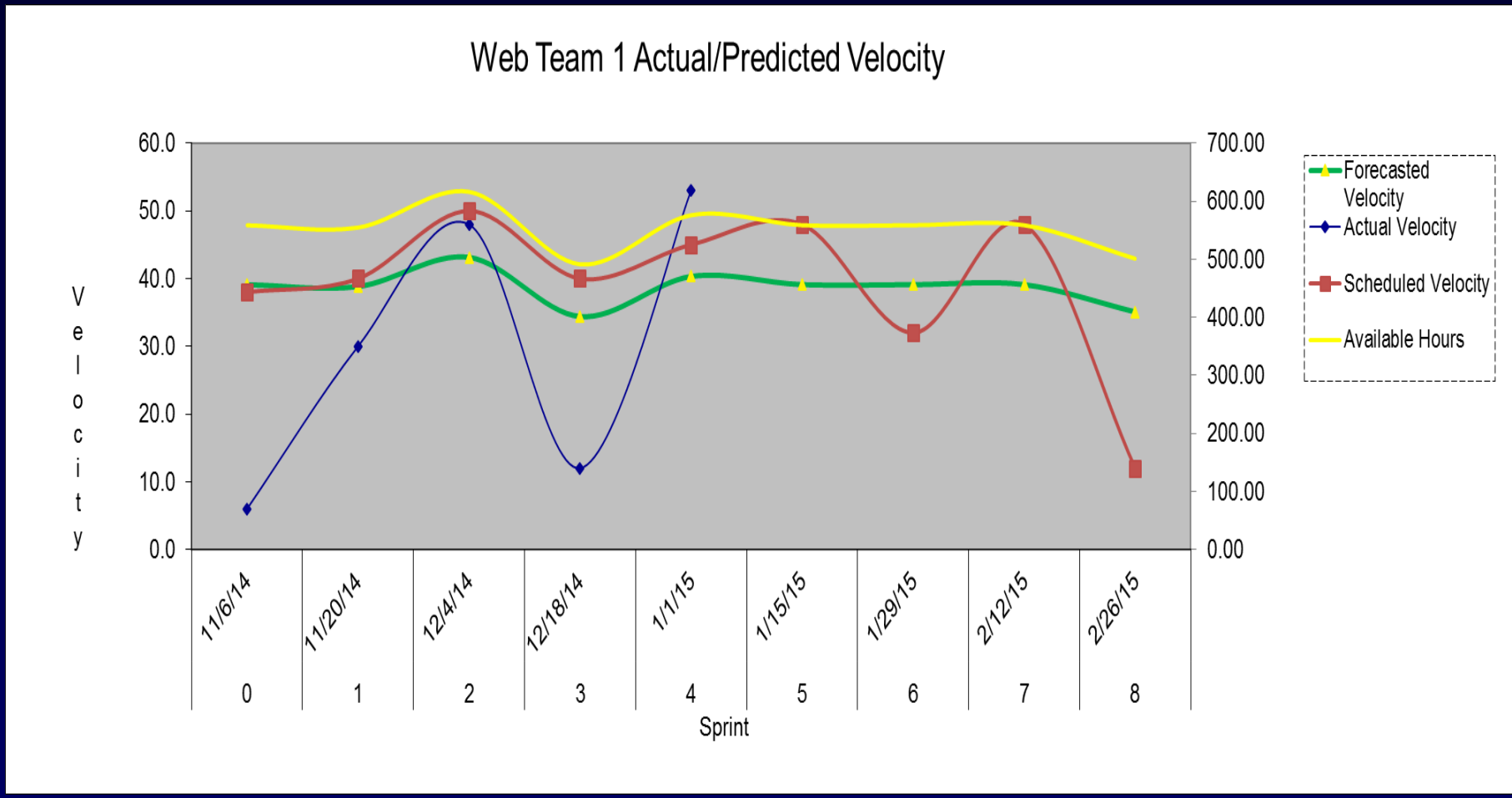
Track Release Plan:

- Record the actual points completed in Actual Points Completed column J
- Note the actual hours to points ratio is computed in the Actual Velocity Factor in column W
- Record “new scope” in the Added Story Points column L which increases the overall total points

Progress In Pictures

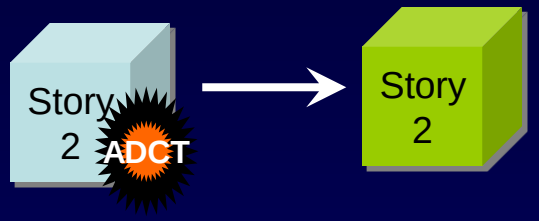


Velocity in Pictures



Related Ideas: Definition of Done

- Is a referee between developers and business to enforce the definition of done
  - Business defines how to measure "done"
  - Developers say it is done
  - Testers verify done
  - Business agrees work is done



- Done is an important concept to velocity - the team does not get "credit" unless the story really is done - we are measured by delivering production ready software

Need for Capacity Planning

- Critical to the release planning meeting (RPM):
  - How many story points can I commit to for a given iteration?
  - Business reps will then allocate stories to the iterations based on this capacity
  - Done ahead of each release planning meeting
  - Also useful for portfolio planning

Team Ratio Assumption

- For every three or so developers:
  - One Product Owner, business person
    - Maybe a business analyst as well?
  - One imbedded tester
  - 1/2 automated function test writer
  - 1/4 build/environment support person
- Double the developers, double the other folks
- Of course, your mileage may vary ☺

Capacity Adjustment Ideas

- Defects - plan a minimum of 10% for defect work - don't allow high and med. defects to accumulate
- Adjust capacity to ensure no defect inventory
- Last two iterations, take 1/2 of capacity for defects
- Holiday months, drop capacity by 10%
- New people should not be planned at 100%, there is ramp up time

Summary

- Basic team information was gathered
- Some assumptions were entered
- A forecast was created for our review
- Adjustments could be made to the plan until we get everyone on the same page
- We have completed our capacity plan as the foundation for a release plan
- Actuals can be entered to see how we are progressing on our plan

Questions

Contact: Shane Hayes, shane @ Octopus – software. com

Copyright / License