NORTHROP GRUMMAN

# Planning, Estimating, and Monitoring Progress in Agile Systems Development Environments

Systems & Software Technology Conference
Salt Lake City
April 2010

Dr. Suzette S. Johnson
Northrop Grumman Corporation
Agile Engineering and Leadership
Suzette.Johnson@ngc.com

# Report Documentation Page

*Form Approved*
*OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

| | | |
|---|---|---|
| 1. REPORT DATE<br>**APR 2010** | 2. REPORT TYPE | 3. DATES COVERED<br>**00-00-2010 to 00-00-2010** |

| | |
|---|---|
| 4. TITLE AND SUBTITLE<br>**Planning, Estimating, and Monitoring Progress in Agile Systems Development Environments** | 5a. CONTRACT NUMBER |
| | 5b. GRANT NUMBER |
| | 5c. PROGRAM ELEMENT NUMBER |
| 6. AUTHOR(S) | 5d. PROJECT NUMBER |
| | 5e. TASK NUMBER |
| | 5f. WORK UNIT NUMBER |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)<br>**Northrop Grumman Corporation,2980 Fairview Park Drive,Falls Church,VA,22042** | 8. PERFORMING ORGANIZATION REPORT NUMBER |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) | 10. SPONSOR/MONITOR’S ACRONYM(S) |
| | 11. SPONSOR/MONITOR’S REPORT NUMBER(S) |

12. DISTRIBUTION/AVAILABILITY STATEMENT
**Approved for public release; distribution unlimited**

13. SUPPLEMENTARY NOTES
**Presented at the 22nd Systems and Software Technology Conference (SSTC), 26-29 April 2010, Salt Lake City, UT. Sponsored in part by the USAF. U.S. Government or Federal Rights License**

14. ABSTRACT

15. SUBJECT TERMS

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT<br>**unclassified** | b. ABSTRACT<br>**unclassified** | c. THIS PAGE<br>**unclassified** | **Same as Report (SAR)** | **31** | |

# Discussion Outline

- Agile Systems Development Introduction
- Scenario
  - Levels of Planning
  - Estimating
  - Monitoring Progress
- Summary
- Recommended Readings and References

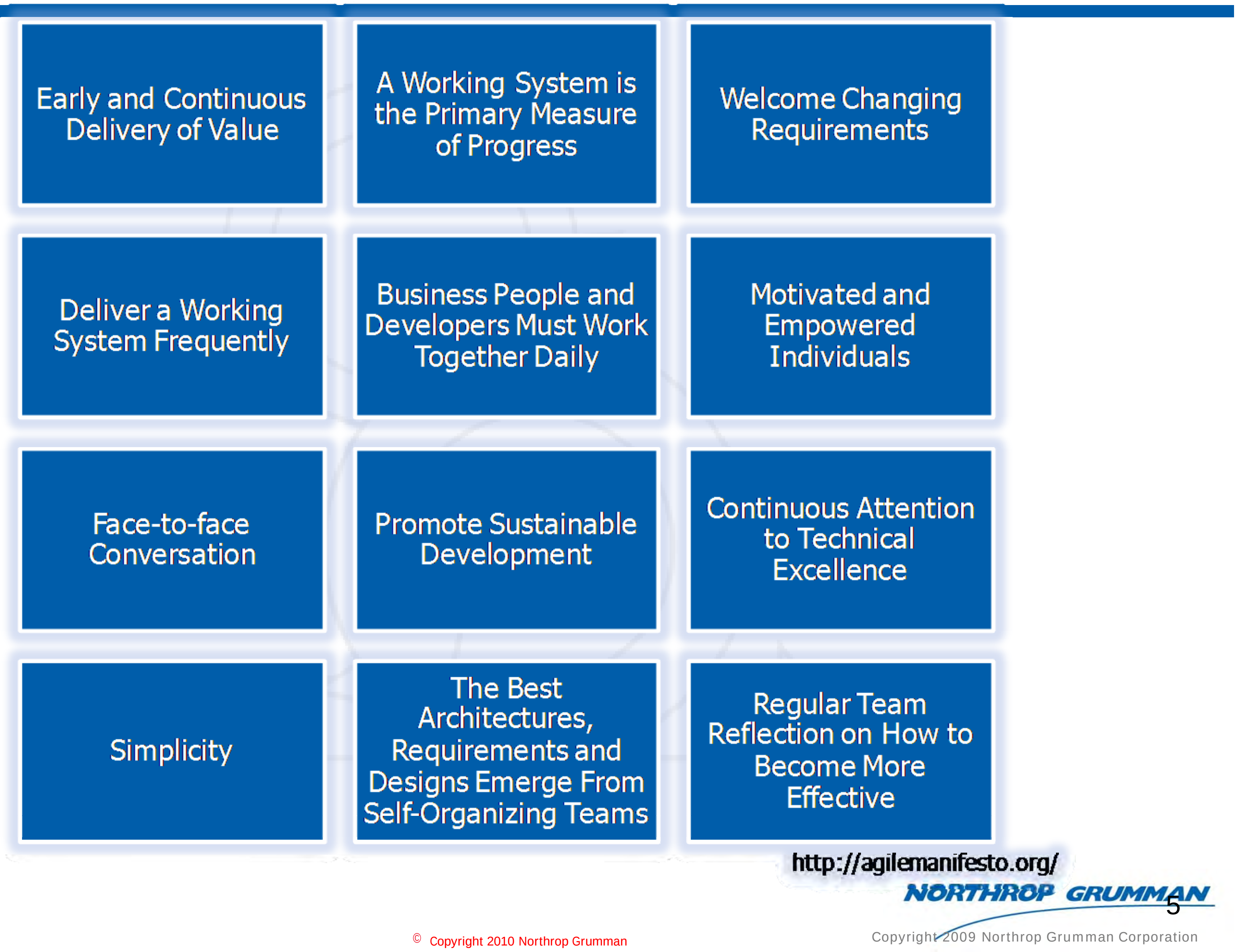
# Agile Systems Development

*Promote rapid delivery of value to customers*

# What is Agile Systems Development?

- Agility is a set of demonstrated industry best practices for developing *software systems*
- Agility focuses on:
  - Principles and Values
  - Inspect and Adapt
  - Constant Commitment to Quality
  - Focus on the Value Stream
  - Team Ownership and Empowerment

# Agile Principles



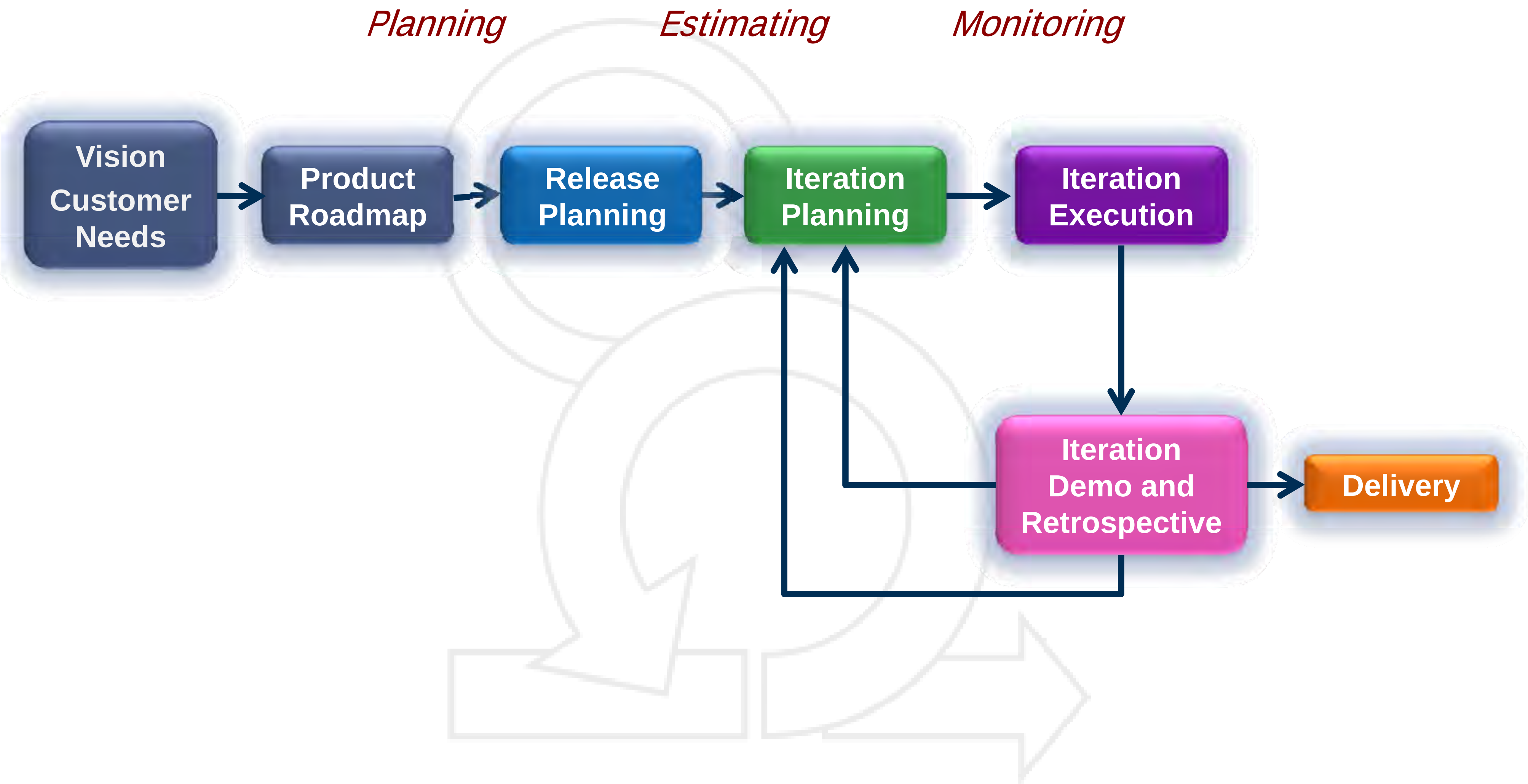
| | | |
|---|---|---|
| Early and Continuous Delivery of Value | A Working System is the Primary Measure of Progress | Welcome Changing Requirements |
| Deliver a Working System Frequently | Business People and Developers Must Work Together Daily | Motivated and Empowered Individuals |
| Face-to-face Conversation | Promote Sustainable Development | Continuous Attention to Technical Excellence |
| Simplicity | The Best Architectures, Requirements and Designs Emerge From Self-Organizing Teams | Regular Team Reflection on How to Become More Effective |

http://agilemanifesto.org/

# Agile Terminology

| Term | Definition |
|---|---|
| Product Backlog | Requirements/User Stories to be completed |
| Iteration | Fixed time-box in which development occurs |
| User Story | Similar to a requirement<br>“As a ***user*** I want to ***action*** so that ***purpose***” |
| Velocity | The number of user story “points” delivered in a iteration |
| Capacity | The hours the development team is available to work within an iteration |
| Product Burn Down Chart | Progress for the release; Focuses on the remaining user story points |
| Iteration Burn Down Chart | A development team’s progress; Focuses on the remaining hours |
| Product Owner | Owns the product backlog, assigns priority to user stories |
| The Team | Cross functional team |

# High Level Agile Stages



*Planning*

*Estimating*

*Monitoring*

Vision Customer Needs

Product Roadmap

Release Planning

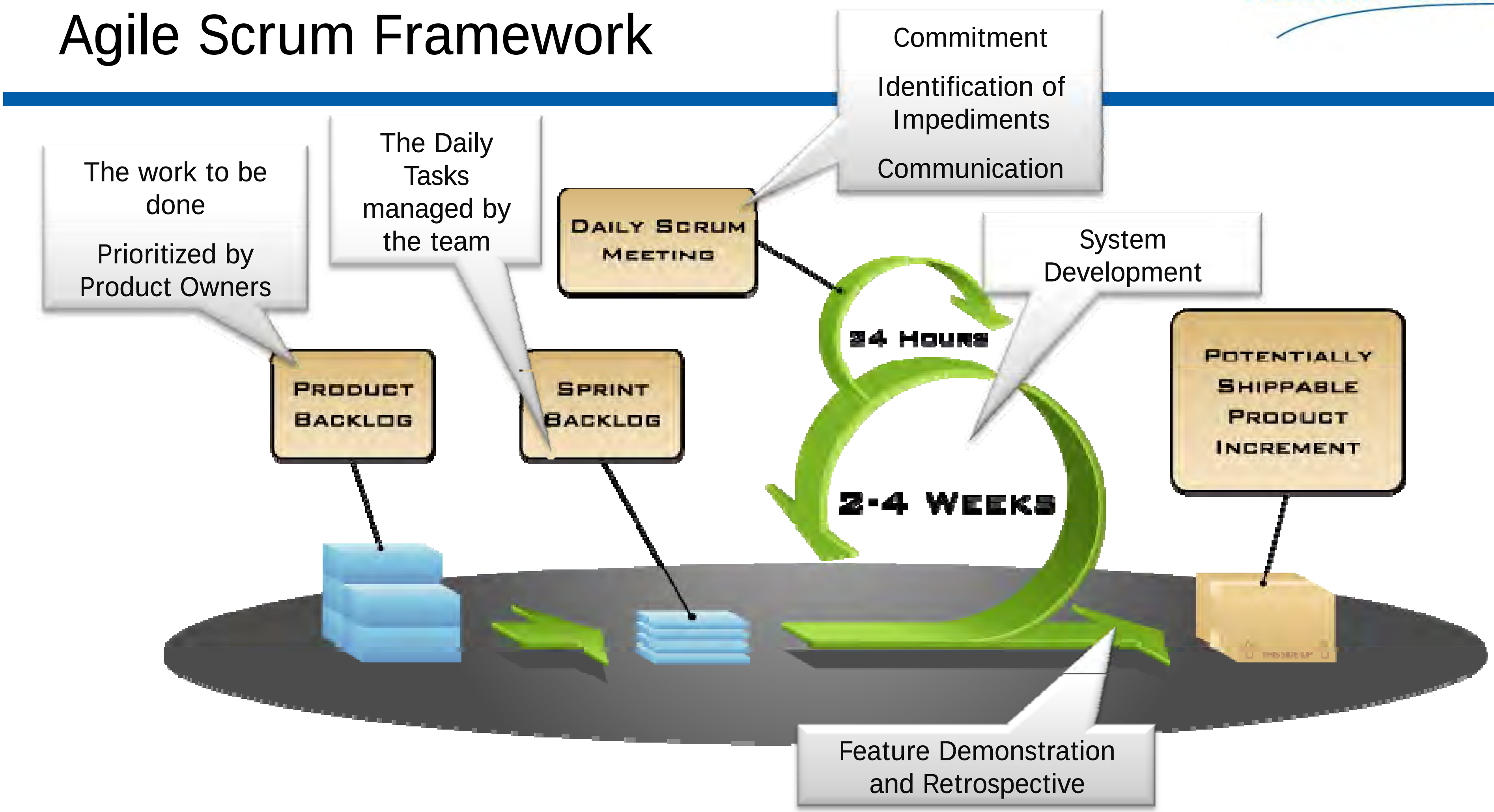
Iteration Planning

Iteration Execution

Iteration Demo and Retrospective

Delivery

# Agile Scrum Framework



The work to be done

Prioritized by Product Owners

The Daily Tasks managed by the team

Commitment

Identification of Impediments

Communication

System Development

Feature Demonstration and Retrospective

An example of an agile management framework

**Inspect and Adapt**

**Visibility and Transparency**

Image from: Mike Cohn, MountainGoatSoftware.com

Reference: Ken Schwaber, www.controlchaos.org

NORTHROP GRUMMAN

# Scenario

## Planning, Estimating, and Monitoring

# Project Scenario: RestEZ

- Duration: One Release
- Product: Development of a hotel website for RestEz
- Size of Project: ~10 persons/team; 10 teams = 100 people
- Release cycle is January 4 – March 31
- Six two-week iterations within the Release
- Planning for the Release is started prior to January 4th
  - Determining high level capabilities for the release
  - Story Writing workshop (Capability to User stories)
- Release Planning meeting is January 4th
- Iteration 1 detailed planning is in the afternoon of January 4th
- January 5th is the first day of development

# Agile Metrics During the Release

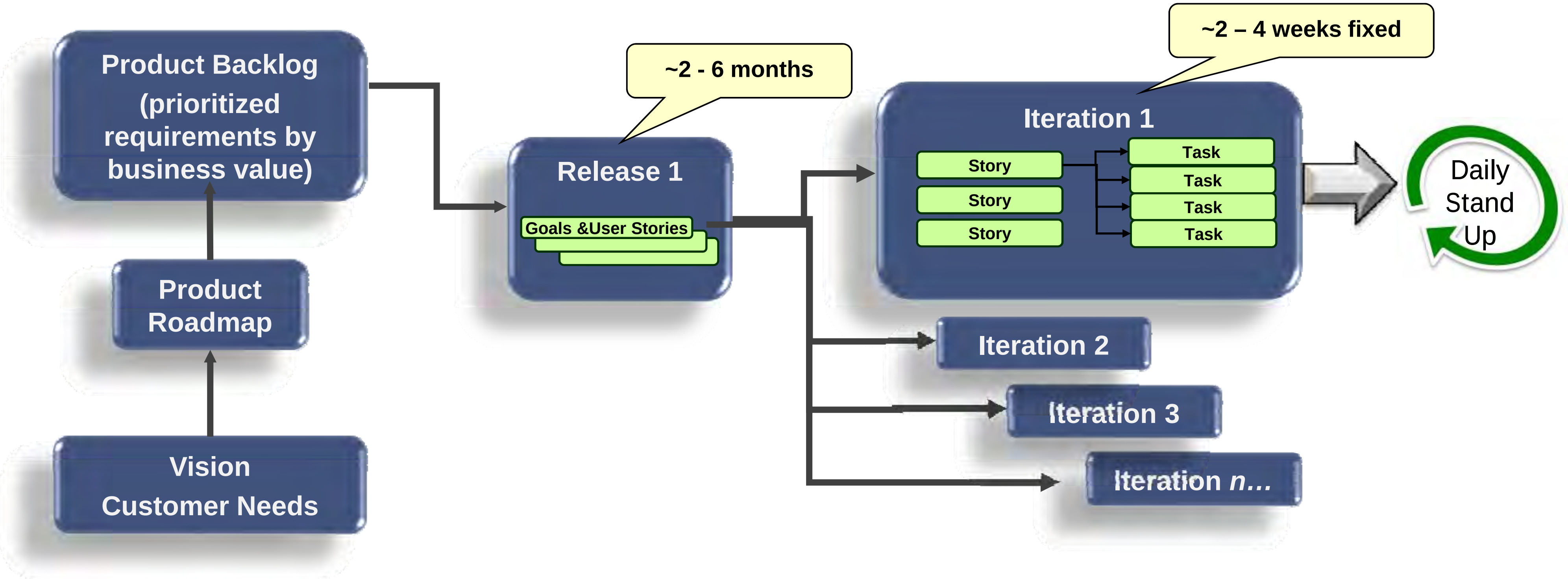
- Planning
  - Velocity
  - Capacity
  - Total story points planned for the release
  - Length of iteration and release
  - Planned work hours
- Monitoring Progress
  - Product Burndown
  - Iteration Burndown
  - User Stories by State
  - Tracking Defects
  - Test Metrics
  - Actual worked hours

Scenario Walkthrough...

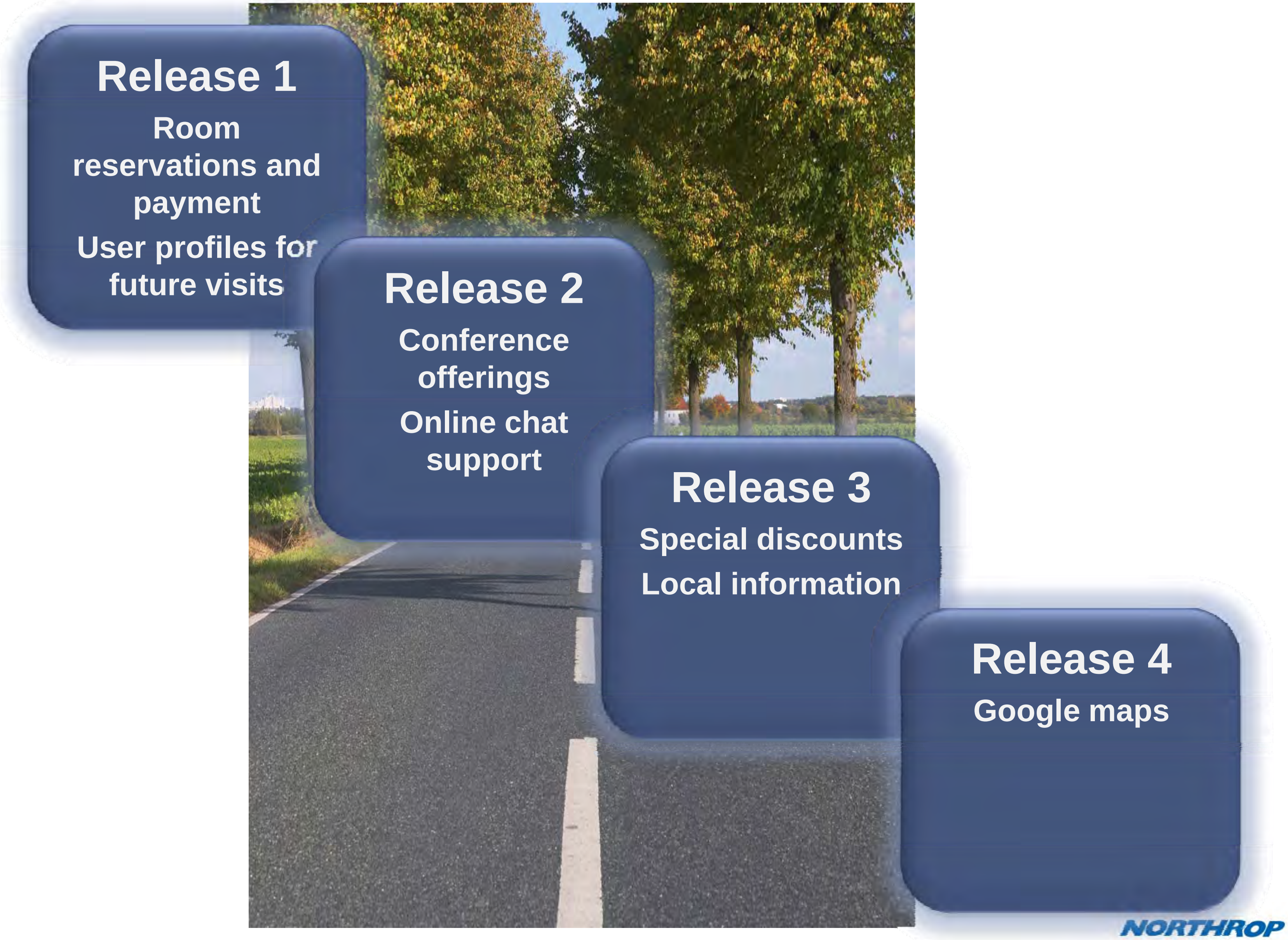
# *Planning and Estimating*

# Levels of Planning

## Vision, Roadmap, Release, Iteration, Daily



# Product Roadmap



Release 1

Room reservations and payment

User profiles for future visits

Release 2

Conference offerings

Online chat support

Release 3

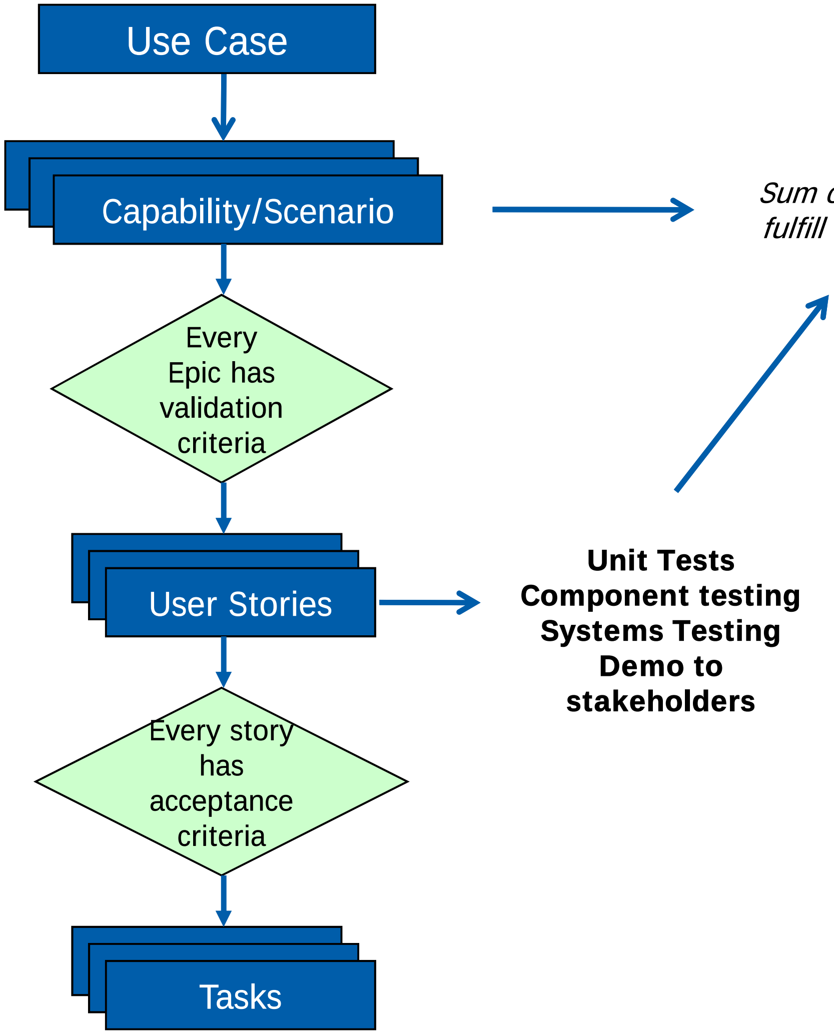
Special discounts

Local information

Release 4

Google maps

# Release Planning: Use Case and User Stories



Use Case

Capability/Scenario

Every Epic has validation criteria

User Stories

Every story has acceptance criteria

Tasks

***Written at the capability level and can be completed in one Release***

*Sum of all the test cases to fulfill the Release level test criteria*

***Written at a level that the team can complete in one iteration***
***Owned by the Product Owner***

**Unit Tests**
**Component testing**
**Systems Testing**
**Demo to stakeholders**

***Tasks written by the Team***
***<=8 hours of work***
***Based on a definition of "done"***
***Owned by the Team***

# User Stories

## What is a User Story?

- Functional stories
  - often based off a scenario of a use case
  - On large projects a user can be another system
- Non-functional stories
- Definition of Done
  - Design, Write tests, code, unit tests, documentation, etc.
- No credit for partial work – either done or not done

## Estimation

- User Story Points
  - Bigness of the task
  - Influenced by (a) how hard it is (b) how much there is to do and (c) amount of uncertainty
  - Estimated by the team
  - Relative values
  - Total story points for the Release

# Roadmap to Release Plan

Example: Hotel Website

## Release 1 Capability 1: Make Room Reservations

Release Plan (User Stories)

*How much can we do?*

| Business Value | User Stories | Test | Points |
|---|---|---|---|
| 80 | As a vacationer, I want to search room availability... | • Test with search on 1 room<br>• Test with search on executive suite.... | 12 |
| 75 | As a vacationer, I want to save my request... | Test<br>Objective | 8 |
| 70 | As a vacationer, I want to pay with a credit card... | Test<br>Objective | 16 |
|  | Story *n*... |  |  |

# Release Plan, Iteration Plan, Daily Plan

Example: Hotel Website

## Capability 1: Make Room Reservations

### Release Plan (User Stories)

| Business Value | | Test | Points |
|---|---|---|---|
| 80 | As a vacationer, I want to search room availability... | Test with search on 1 room<br>Test with search on executive suite.... | 12 |
| 75 | As a vacationer, I want to save my request... | Test<br>Objective | 8 |
| 70 | As a vacationer, I want to pay with a credit card... | Test<br>Objective | 16 |

### Iteration Plan (Tasks)

| | Hours |
|---|---|
| Design Review | 4 |
| Install Baseline | 4 |
| ICD Updates | 8 |
| Acquire Test Data | 8 |
| Code | 24 |
| Develop Tests | 8 |
| Run Tests | 8 |

### The Daily Plan

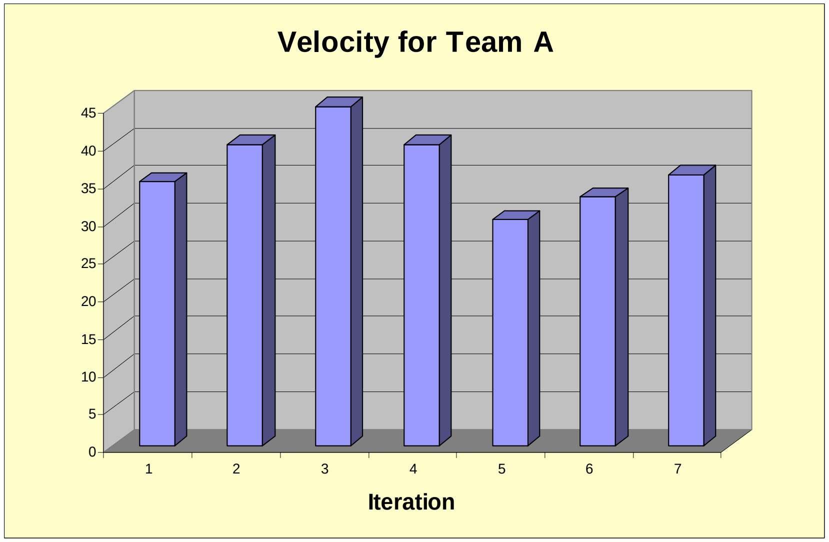
Yesterday I started on the interface….

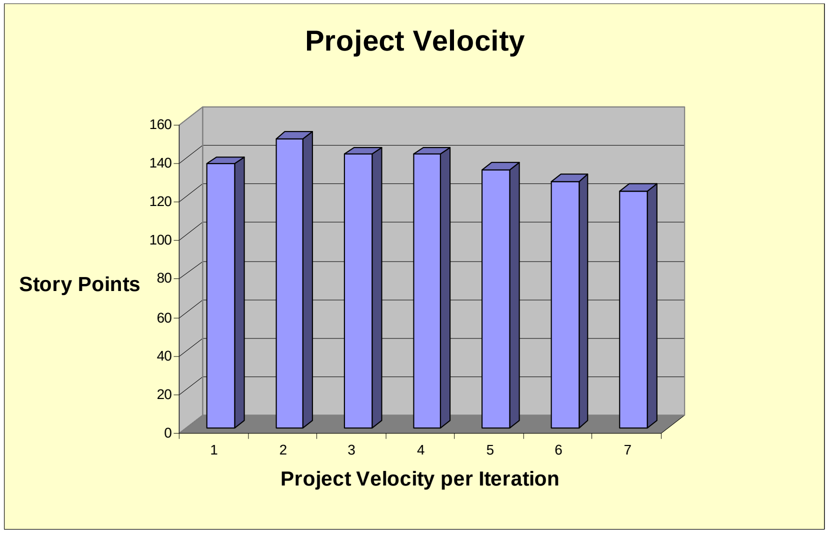
Today I plan to…

The one thing standing in my way…

# Velocity (Based on history)

- Velocity is the amount of work a development team completes in an iteration (story points completed)
- Velocity is a range; Look for the high, the low, and the mean.





Team A Velocity

High: 45 story points

Low: 30 story points

Mean: 37 story points

Project Velocity per Iteration

High: 155 story points

Low: 120 story points

Mean: 137 story points

# Determining Team Capacity for an Iteration

- Capacity is the development team members' available hours to work per iteration
- Revisited each iteration
- Compare planned hours versus actual hours
- Compare team capacity hours to the hours in the iteration

Example for a two-week iteration

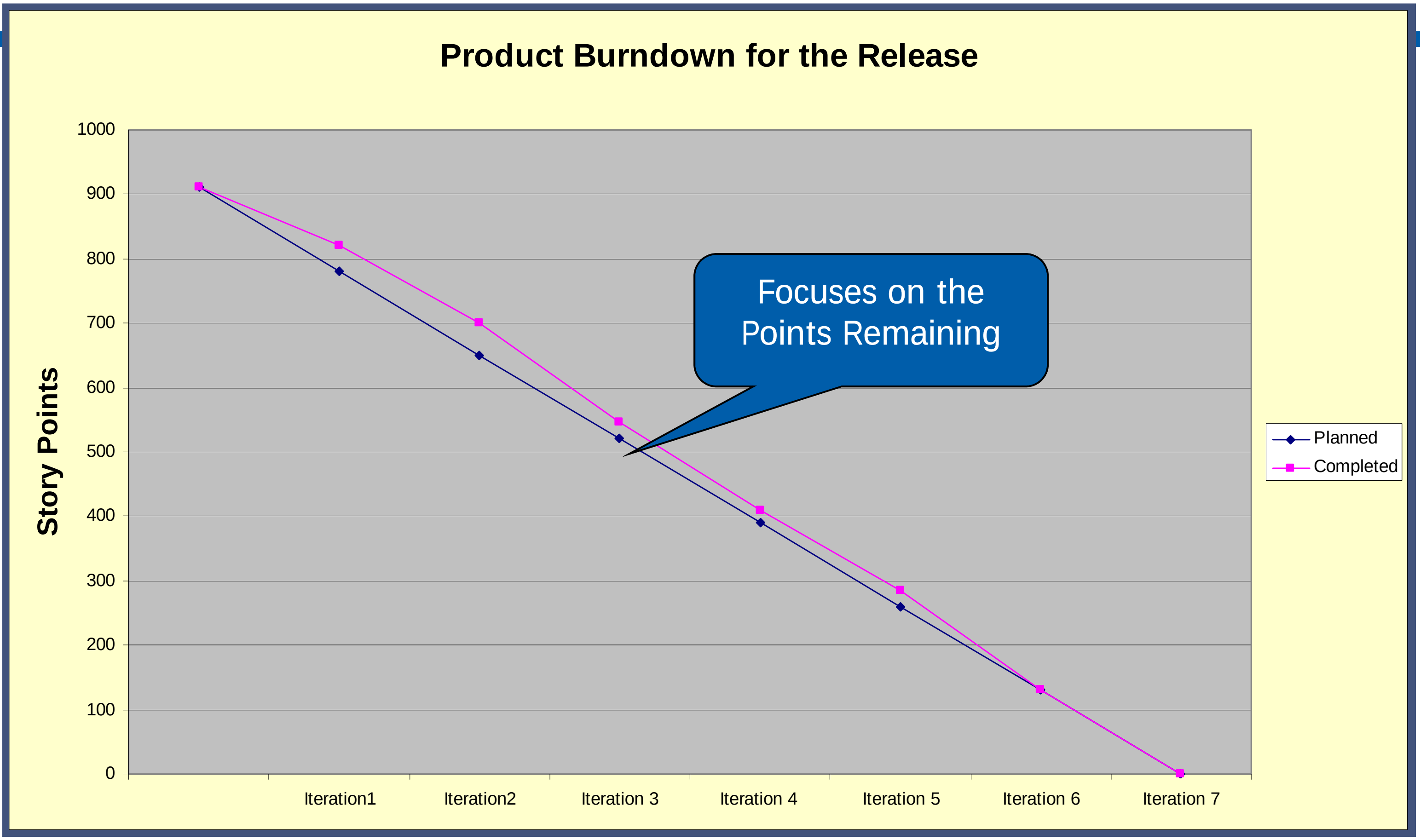
| Team Member | Hours per day | Total hours per iteration |
|---|---|---|
| Bill | 5 | 50 |
| Scott | 5 | 50 |
| Chris | 8 | 80 |
| Andy | 7 | 70 |
| Cindy | 7 | 70 |
| Mike | 8 | 80 |
| TEAM TOTAL | 40 | 400 |

# The Release Plan

- What is in the release plan?
  - Capabilities identified
  - User stories (functional and non-functional requirements)
    - Story points and prioritized
    - Project Teams average about 137 user story points per iteration; for a release with 6 iterations this is about 900 story points. The scope is 720-930 user story points of work.
  - Total number of user stories planned (125)
  - Total number user story points planned (~910 user story points)
  - Known or assumed velocity by development team and project team
  - Planned hours (WBS element)

# *Monitoring Progress*

# RestEZ: Product Burndown for the Release



Focuses on the Points Remaining

- 910 points planned
- Project team or development team perspective
- Progress made and work remaining
- Based on story points

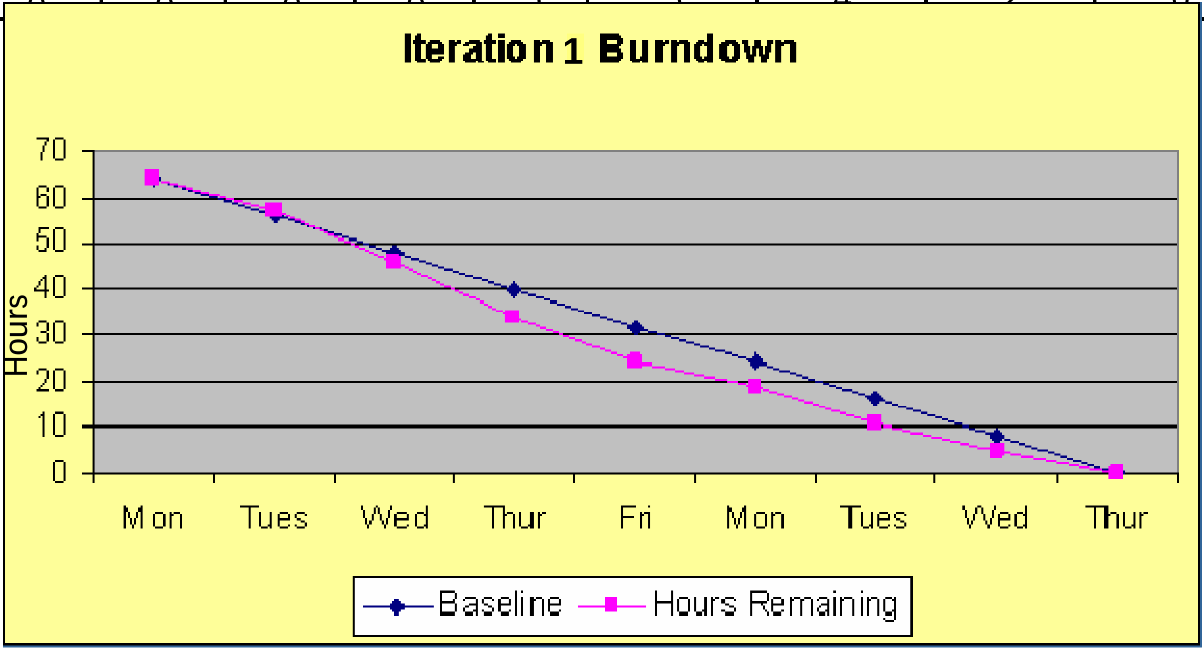
- Initiates discussion
- Reviewed every iteration
- Can be a “Burnup” chart
- Reports user stories completed against the plan for the release

# A Development Team’s Iteration Burndown

User Story: As a vacationer, I want to search room availability...

| Iteration Tasks | Owner | Status | Mon | Tues | Wed | Thur | Fri | Mon | Tues | Wed | Thur |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Design Review | Scott | Completed | 4 | 4 | 4 | 3 | 2 | 0 | 0 | 0 | 0 |
| Install baseline | Bill | Completed | 4 | 4 | 4 | 0 | 0 | 0 | 0 | 0 | 0 |
| ICD updates | Scott | Completed | 8 | 8 | 6 | 3 | 3 | 2 | 0 | 0 | 0 |
| Acquire test data | Bill | Completed | 8 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Code | Scott | Completed | 24 | 20 | 16 | 14 | 10 | 10 | 3 | 0 | 0 |
| Develop tests | Scott | Completed | 8 | 8 | 8 | 6 | 4 | 4 | 4 | 3 | 0 |
| Run Tests | Scott | Completed | 8 | 8 | 8 | 8 | 5 | 3 | 4 | 2 | 0 |

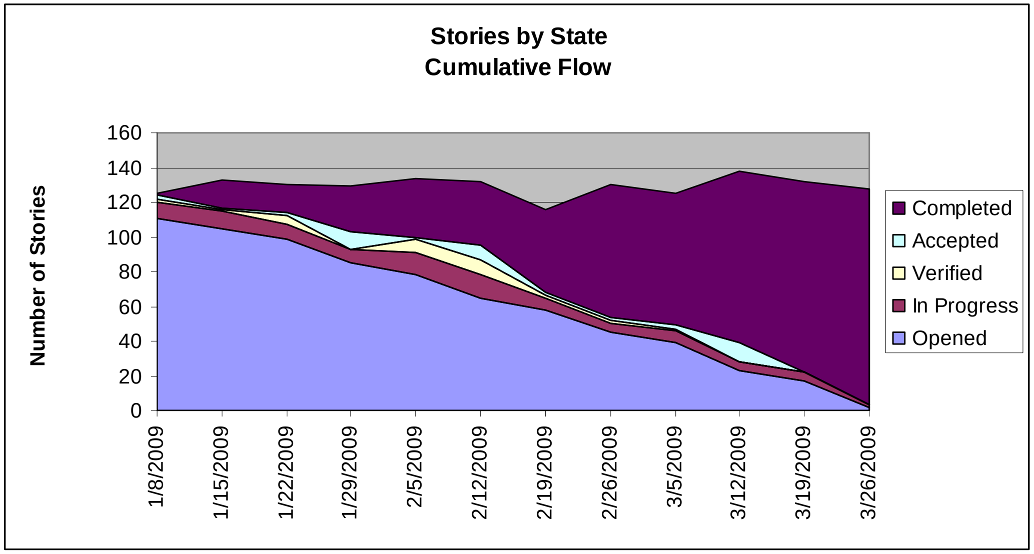
- Emphasis on effort in hours
- Based on task hours for each story
- Hours remaining
- Updated daily
- Reviewed daily



**The Iteration Burndown is for the Team**

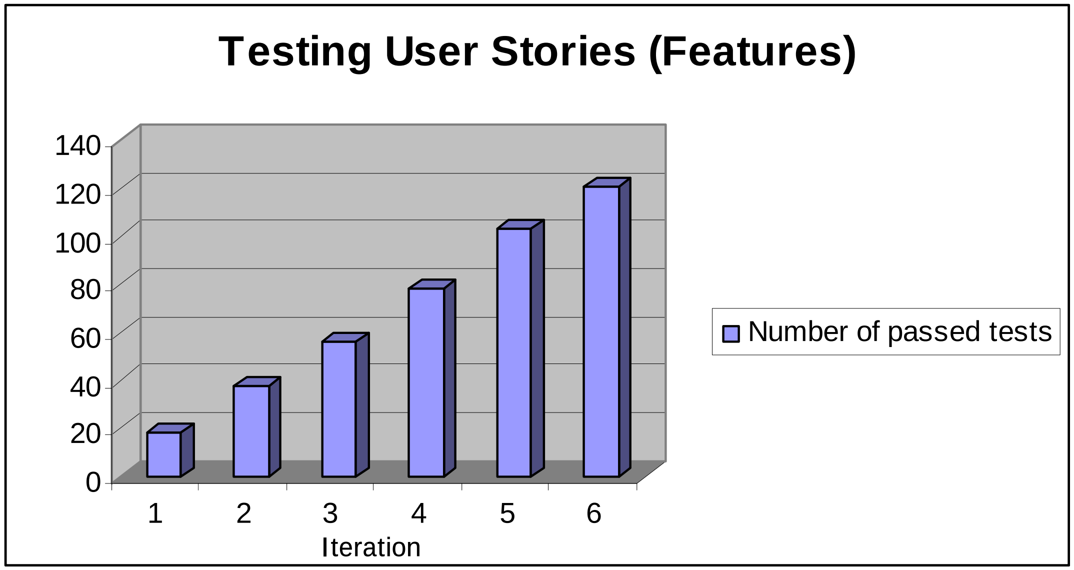
# Tracking Stories by State

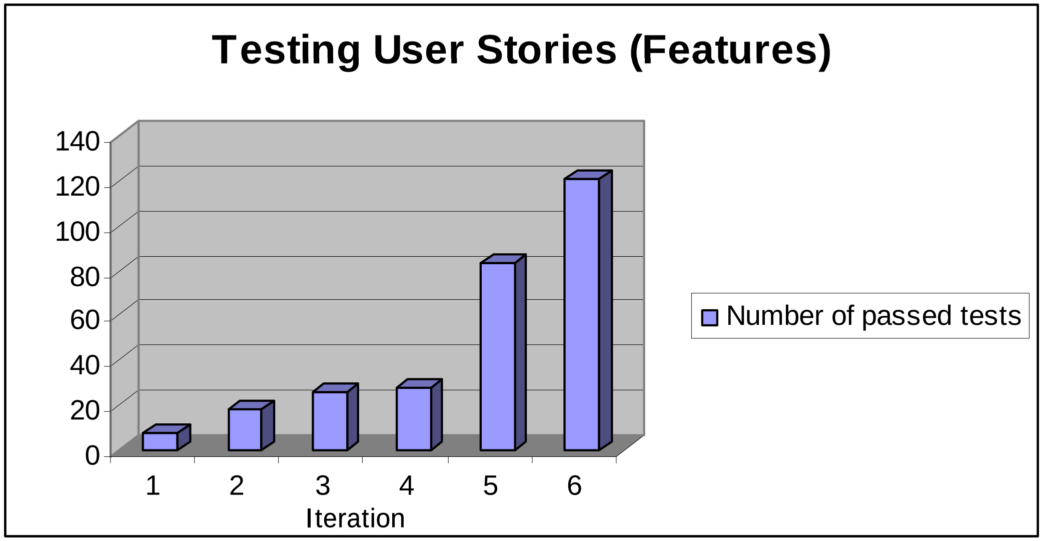
- How work is progressing during the iteration
- State examples: Opened, In Progress, Verified, Accepted, and Completed
- Based on total number of stories for the release
- May help identify bottlenecks
- *Verified* means all tests have passed on the system
- *Completed* means it is ready for release (i.e. potentially shippable)



Reference and for additional information: http://leanandkanban.wordpress.com/2009/04/04/useful-agile-metrics-cumulative-flow/

# Testing the User Stories





Compare the two graphs. Graph 1 indicates a relatively constant increase in “passed” user stories.

What about graph 2?

Metrics can help teams identify bottlenecks and other challenges on the project.

What is the challenge in graph 2?

# Defect Metrics

- Defect metrics include:
  - Number of open defects at the end of each iteration
  - Number of defects found during systems integration
  - Number of defects found in production
  - The type of defects occurring
    - Useful in determining root cause
- May give indication as to what processes need to change
- Agile teams strive for “zero” defects
  - Part of the Definition of Done is “no critical defects against the story” or the story isn’t “done”

# Summary

- Agile systems development environments:
  - Emphasize ongoing iterative development with completed, demonstrable functionality at the end of every iteration
  - Embrace practices that support changing requirements and mission needs
  - Engage in multiple levels of planning
- Valuable measures for estimating and monitoring progress:
  - Velocity (scope)
  - User stories planned versus user stories completed (progress)
  - User stories added to a release; not part of the original plan
  - Task hour burndown (daily progress)
  - Tests passed (what’s working)
  - Hours planned and hours worked

# Recommended Reading and References

| | |
|---|---|
| ***Creating Adaptive Businesses*** | |
| Adaptive Enterprise | Steven Haeckel |
| Complexity leadership theory: Shifting leadership from the industrial age to the knowledge era | Marion, McKelvey, & Uhl-Bien. (2007). *Leadership Quarterly, 18*(4), 298-318. |
| ***Agile Development Practices*** | |
| Agile Project Management with Scrum | Ken Schwaber |
| Agile Software Development: Adaptive systems principles and best practices | Meso and Jain |
| Agile Software Development with Scrum | Ken Schwaber and Mike Beedle |
| Agile Testing | Lisa Crispin and Janet Gregory |
| Implementing Lean Software Development | Poppendeick |
| ***Metrics*** | |
| Agile EVM – Earned Value Management in Scrum Projects | Sulaiman, Barton, Blackburn |
| Agile Metrics (Agile 2009 Conference proceedings) | Dan Rawsthorne, Danube |

NORTHROP GRUMMAN

# The Need for Change

- Shortened product life cycle and technological advancements
- Shortened development times
- Decreased time-to-market
- Complexity of systems
- Large program failures
- Desire for improved transparency of progress
- Reduce risk