

# Agile Measurement Workshop

## Software Development Metrics

July 29, 2010

Dan Beisel

[Dan.Beisel@ngc.com](mailto:Dan.Beisel@ngc.com)

Master Planner in Agile and Traditional methods

# Today's Outcomes

- Overview of agile software (SW) development
  - Establish a common vocabulary
- Measures throughout the agile lifecycle
- High Level Comparison of Agile to traditional measures
- Hands on experience
- New Indicators



Goal: Introduction to agile SW development & metrics

# Agenda

- Introductions
- Agile life cycle overview
  - Establish a common vocabulary, metrics overview, break
- Planning measures
  - Stories, estimation, exercise
- Execution metrics & monitoring
  - Burn down, velocity, exercise, break
- Retrospectives
  - Impounds, predicting next iteration
- ICM Review / Update Recommendations
- Closing comments



Goal: Introduction to agile SW development & metrics

# Introductions

- Introduce yourself
  - Name / Nickname
  - Agile experience (None, Low, Medium, High)
  - Measures experience (None, Low, Medium, High)
  - Particular wants from today
- Select teams for exercises
  - 7 +/- 2 people



Get to know each other for the afternoon

# Agenda

- Introductions
- Agile life cycle overview
  - Establish a common vocabulary, metrics overview, break
- Planning measures
  - Stories, estimation, exercise
- Execution metrics & monitoring
  - Burn down, velocity, exercise, break
- Retrospectives
  - Impounds, predicting next iteration
- ICM Review / Update Recommendations
- Closing comments



Goal: Introduction to agile SW development & metrics

# The Need for Change

## Predictive Versus Adaptive Business Model



### *Industrial Age*

**Repeatable and Predictable**



### *Knowledge Age*

**Inspect and Adapt**

Agile expects & manages changing requirements

## What is Agile Engineering?

- Includes the entire product life cycle
- Impacts the entire organization
- Inspects and adapts
- Focuses on the value stream



## Why Agile Practices?

- Quick reaction capabilities
- Adapt to change
- Shortened product life cycle
- New technological advancements
- Improved transparency of progress and end-to-end accountability and ownership

Agile expects & manages, versus controlling, changing requirements



# 12 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

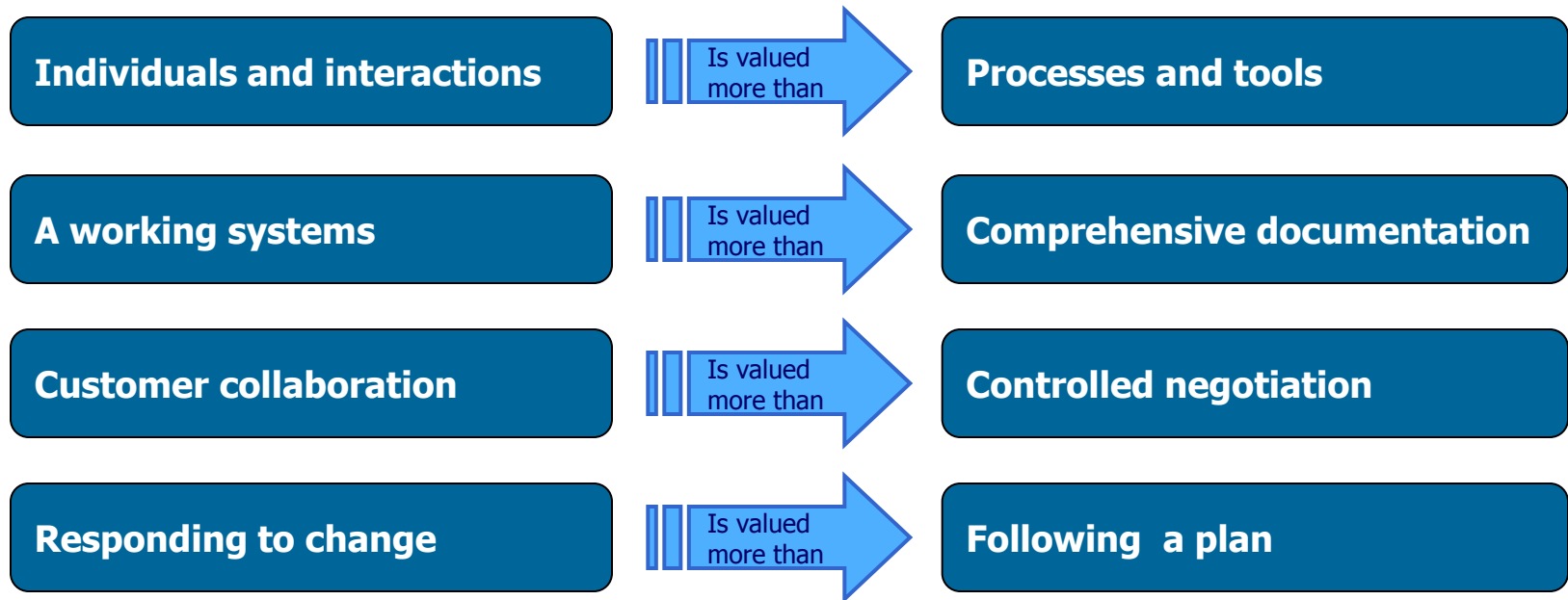
The Best  
Architectures,  
Requirements and  
Designs Emerge From  
Self-Organizing Teams

Regular Team  
Reflection on How to  
Become More  
Effective

<http://agilemanifesto.org/>

Agile principles drive agile practices





<http://agilemanifesto.org/>

That is, while *there is value in the items on the right,*  
we value the items on the left more

Term	Definition
Burndown	The concept, often shown as a graph over time, of working off or “earning” story points toward iteration or delivery completion. Burn down is analogous to velocity (below) in that each measures progress in working off story points toward final delivery.
ESLOC	Effective Source Lines Of Code (ESLOC) is common between agile, traditional methods. Line of code count is a classic SIZE metric
Story (User Story)	A very high-level definition of a requirement, containing just enough information so that the developers can produce a reasonable estimate of the effort to implement it. “As a <b>user</b> I want <b>what</b> so that <b>purpose</b> ”
Story Points	A relative measure of story complexity. An integer established during Planning Poker
Team Capacity	The number of story points a team is capable of delivering in a certain amount of time, usually an iteration. Often expressed with a % Confidence value.
Velocity	The amount of work done over a period of time. Specifically, the daily arithmetic mean of points earned per work day.

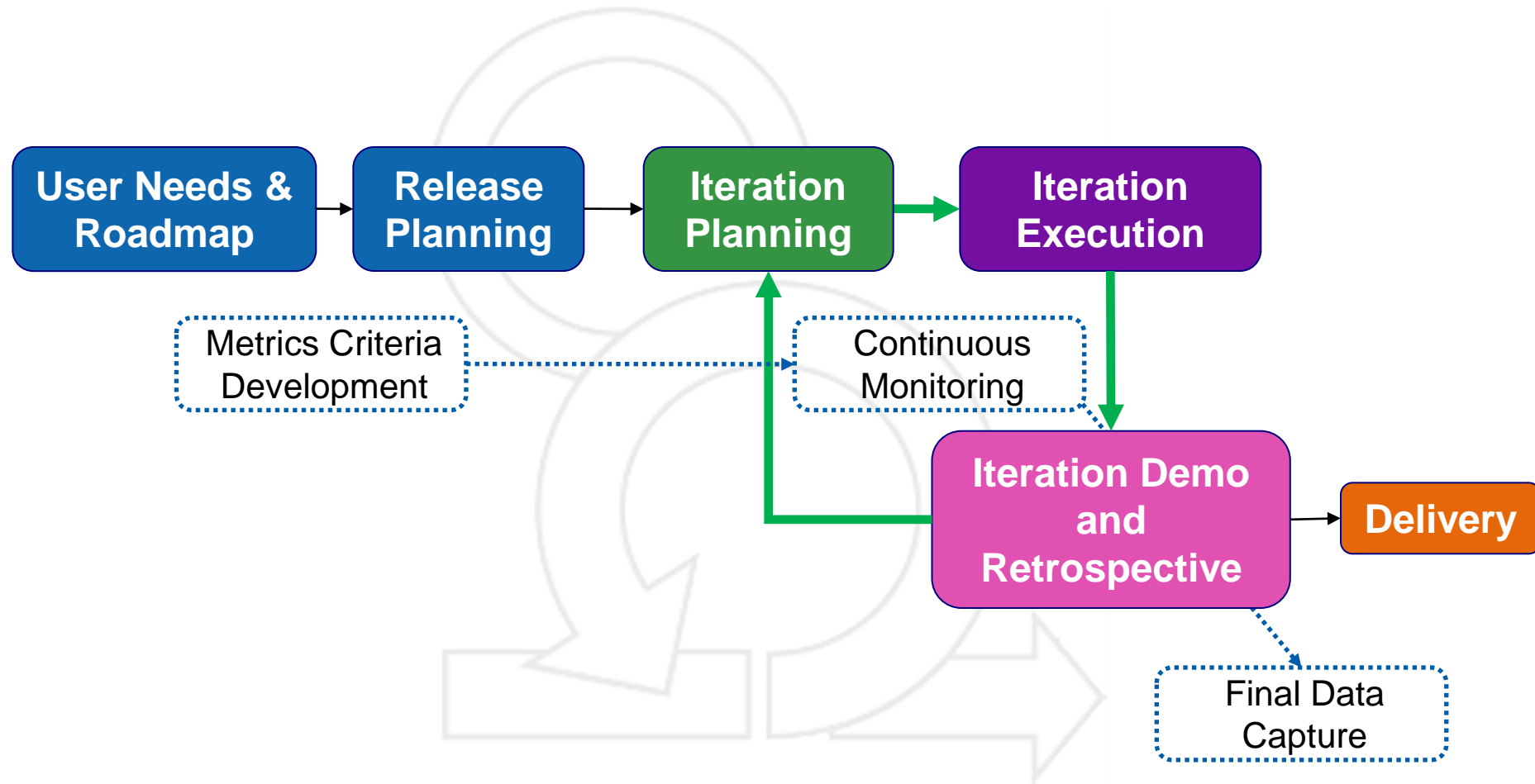
Establishing a common vocabulary is often the hardest step

# Agile Process Vocabulary

This Term...	Definition
Iteration (aka Sprint)	Fixed time-box in which development occurs
Product Backlog	Requirements/User Stories to be completed
Product Owner	Owns the product backlog, assigns priority to user stories Is or represents the customers
Refactor	Agile concept of rewriting software to increase readability or maintainability but never adding or removing capabilities
Release (aka Delivery)	Usually a 2 – 6 month timeframe; formal committed delivery of product
Retrospective	Lightweight, end of Iteration improvement meeting
Scrum Master	Helps the agile team through the process and removes impediments
The Team	Cross functional team

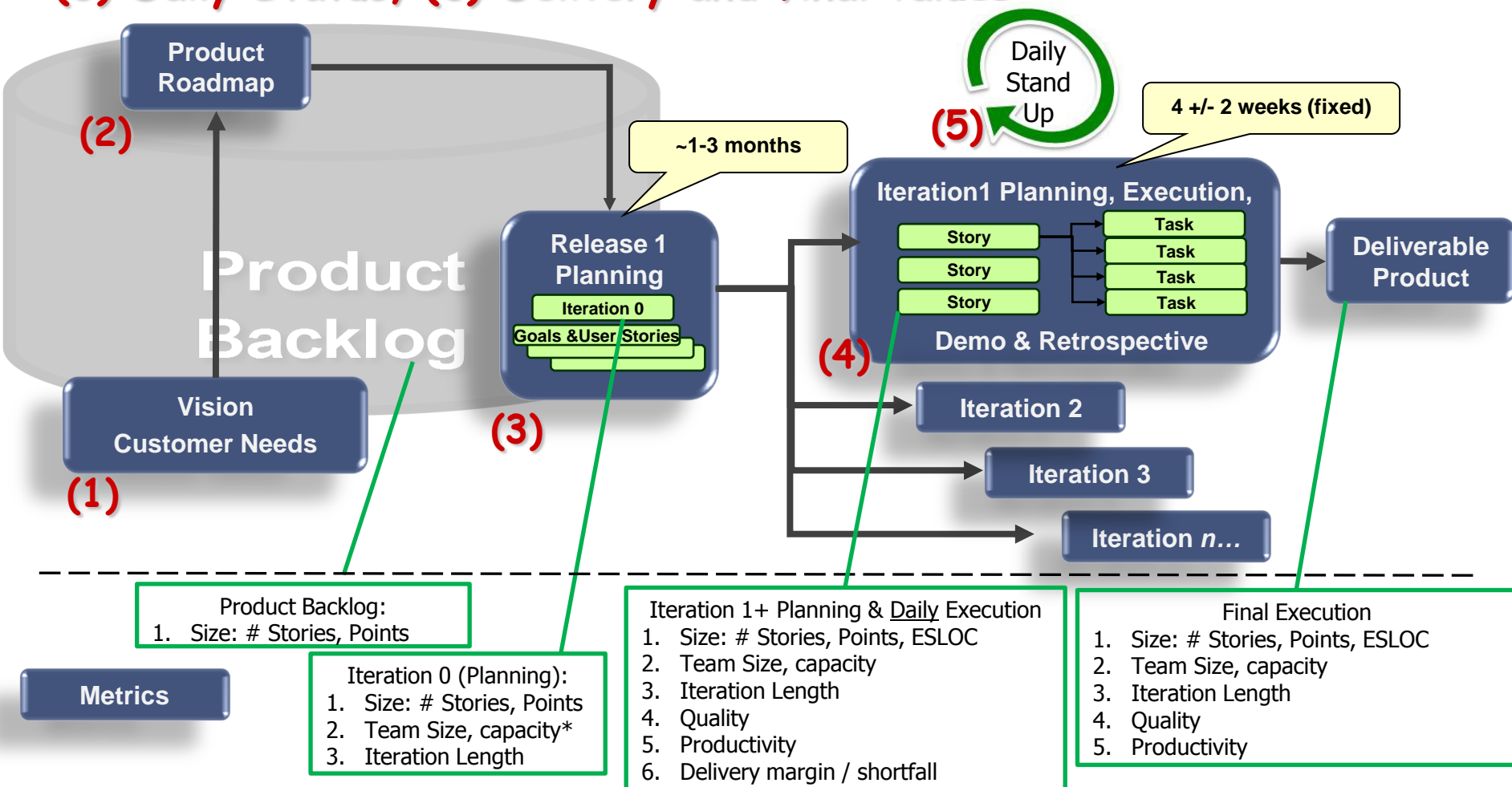
Establishing a common vocabulary is often the hardest step

# High Level Agile Stages



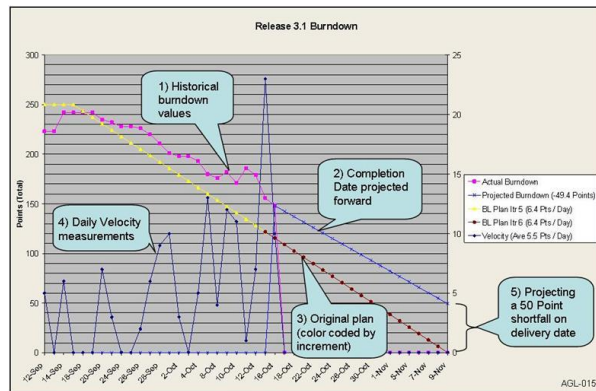
# Agile process & metrics overview comparison

(1) Vision, (2) Product Roadmap, (3) Release, (4) Iteration, (5) Daily Status, (6) Delivery and final values



## Size

- POINTS. Hours vs. Criteria (ADP Fig. 6.5-3)
- VELOCITY = Points / Day. Team Capacity
- ESLOC For comparison to non agile programs - "How are we doing?"



## Color Codes

- Used for dashboards & metrics meetings, PMR
- Tight tolerances give earliest possible indication of trouble, the most time to react

## Cost & Productivity

- Cost – classic HOURS and DOLLARS from BusOps
- Productivity – new measure using POINTS
  - Points / Hour (recommended derived metric)
    - Increase is better
    - Measure daily to get gauge variation
    - Chose a useful denominator
      - Points/100 hours - easier to use values (e.g., 7.1).
      - Points/80 hours – normalized to an pay-period or iteration.

Operational Definition

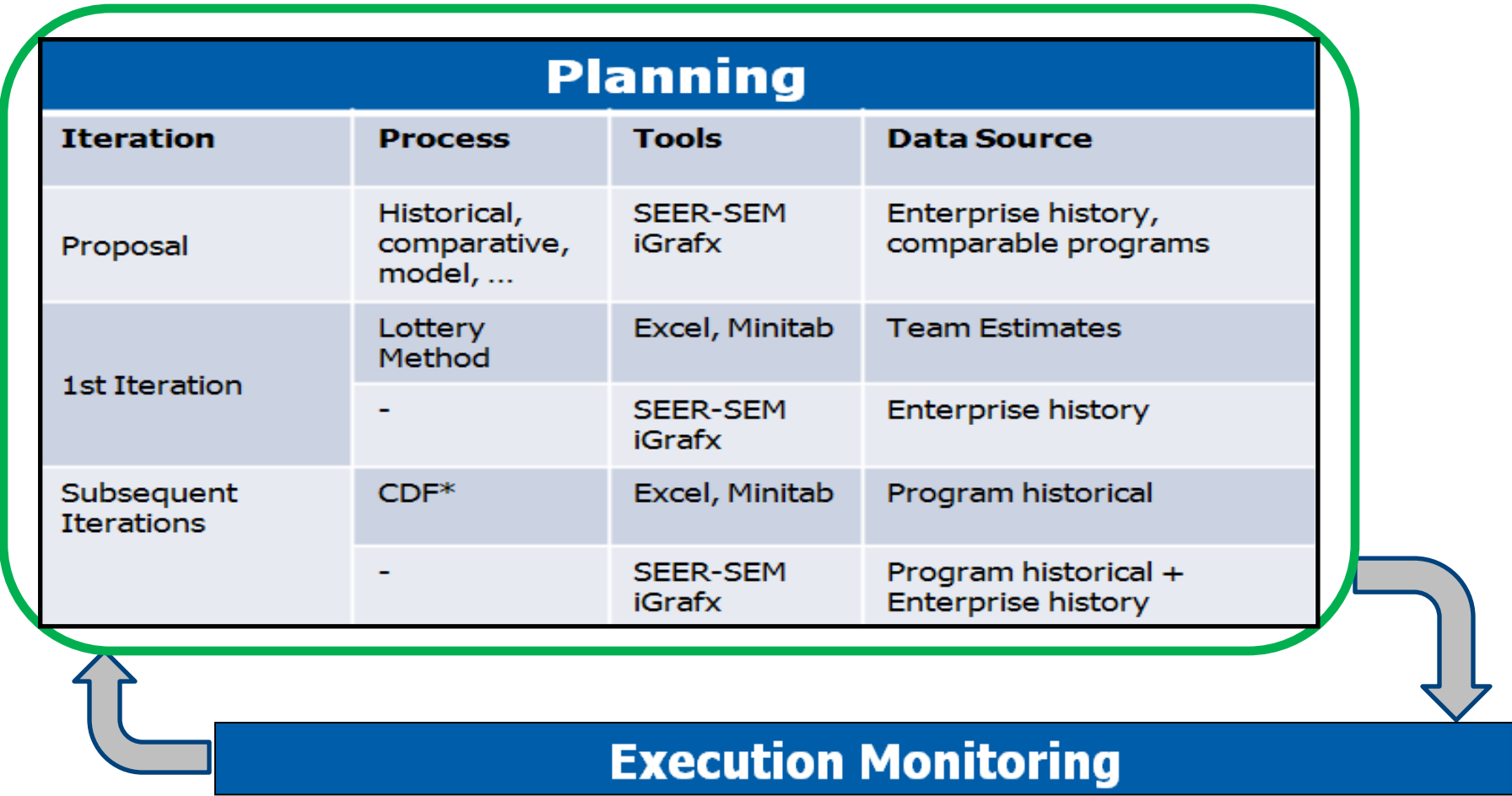
Productivity is key metric to turn resources into capabilities

## Quality & Volatility

- Defects – both Peer Review & traditional SW
  - Extend Peer reviews by points.
- Volatility extend by "Discovery Stories"
  - Stories "discovered" after iteration start
  - Result in unplanned work

Classic measures recast in terms of agile development

# Metrics – Planning to Execution relationship



Planning			
Iteration	Process	Tools	Data Source
Proposal	Historical, comparative, model, ...	SEER-SEM iGrafx	Enterprise history, comparable programs
1st Iteration	Lottery Method	Excel, Minitab	Team Estimates
	-	SEER-SEM iGrafx	Enterprise history
Subsequent Iterations	CDF*	Excel, Minitab	Program historical
	-	SEER-SEM iGrafx	Program historical + Enterprise history

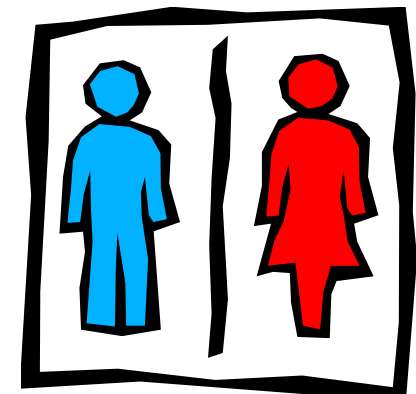
## Execution Monitoring

\* = Cumulative Distribution Function

Metrics close the loop between execution and planning



# Break



Time: 15 minutes

# Agenda

- Introductions
- Agile life cycle overview
  - Establish a common vocabulary, metrics overview, break
- Planning measures
  - Story Definition & Example
  - Story Points Definition
  - Estimation Criteria
  - Exercise
- Execution metrics & monitoring
  - Burn down, velocity, exercise, break
- Retrospectives
  - Impounds, predicting next iteration
- ICM Review / Update Recommendations
- Closing comments



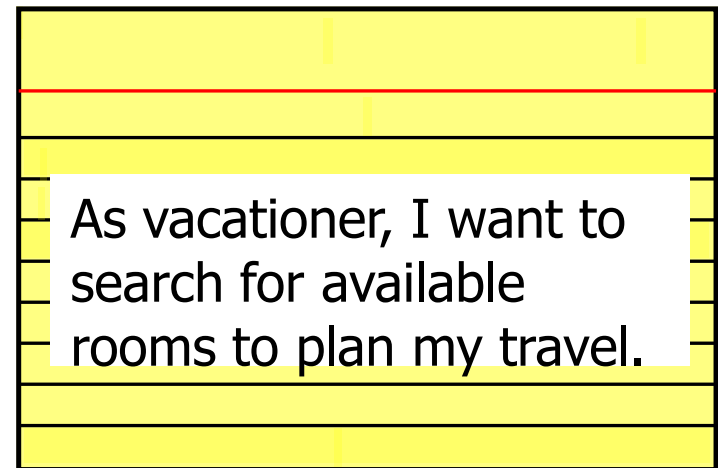
Goal: Introduction to agile SW development & metrics

## 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

## Size Estimation (Story Points)

- Relative integer values
- Considers: effort, complexity
- Consensus of team
- Criteria based (preferred) or hours based (most common)



Stories are the agile version of requirements

# A user story is composed of:

- *Written description* of the story, used for planning and as a reminder
- *Conversations* about the story that serve to flesh out the details of the story
- *Tests* that convey and document details that can be used to determine when a story is complete
  - [http://www.mountaingoatsoftware.com/article\\_view/27](http://www.mountaingoatsoftware.com/article_view/27)

Canonical form:

As a <role> I want to  
<action> because  
<business reason>

# Story Types- Operational Definition

Story Type	Description
New Capability	Classic story representing new business value to the customer
Defect - Formal	Formal defect stories written during formal test or when discovered on the operational system
Defect - Informal	Informal defect stories written prior to formal test as a tracking mechanism; these defects have not yet made it to the operational system
Discovery	New capabilities “discovered” after story writing, usually during the iteration
Documentation	Typically end-user documentation or other customer-required documents
Refactor	Agile concept of rewriting software to increase readability or maintainability but never adding or removing capabilities

Story types stratify metrics for subsequent analysis

# Story Point - Operational Definition

Name (ID)	Size – Story Points
Purpose	Quantification of a story's scope (preferred) or effort (more common)
Description	A relative measure of story complexity
Base measures	Integers captured during Planning Poker. See following slides for "patterns" and scope vs. effort (hours) discussion.
Computation method	Result of Planning Poker process.
Analysis Performed	<ol style="list-style-type: none"><li>1. Consistency across iterations.</li><li>2. Used as base measure in other derived metrics, e.g.:<ol style="list-style-type: none"><li>a. Velocity calculations</li><li>b. Delivery margin / shortfall calculations</li></ol></li></ol>

Story Points are the key base agile metric

# Story Estimation - Hours Based

- Produce estimates in “ideal programmer days”
  - Inherently risky, ideal days never happen.
  - Usually produce lognormal distribution estimates, “skewed to the left” ;
    - Usually underestimate
    - Occasionally overestimate
  - Normalize upper limit if iteration lengths vary
    - e.g. 9 points for 2 week iterations = 18 points for 4 week iterations
  - Estimate effort now, validate every iteration.

Point value patterns		
Pattern	Example	Comments
Fibonacci	1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144...	<ul style="list-style-type: none"><li>• Reflects uncertainty in larger estimates</li><li>• Hard to envision a story as complex as the previous two values added together</li></ul>
Doubling	1, 2, 4, 8, 16, 32, 64	Easiest to envision a story 2x as hard as the previous value

Most common but subjective



# Story Estimation – Criteria Based

- Estimate effort now, derive a relationship later
  - Regression equation is the gold standard goal.
- Other categories to consider:
  - Interfaces
  - Algorithms
  - Human machine interface (HMI)
  - Architecture
  - Testing (Test tools, test data generation, analysis complexity, ...)

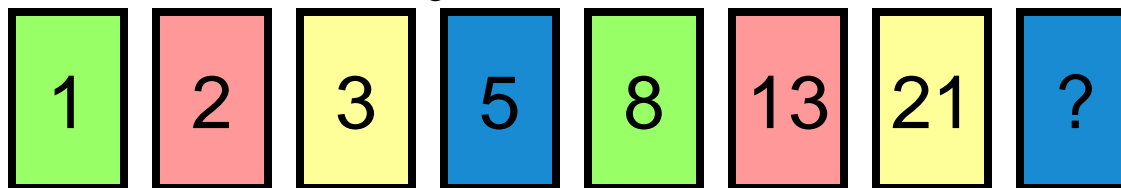
Add as many columns as your enterprise needs

Example Criteria			
Points	Complexity	SW Criteria	(e.g. interface criteria, HMI, ...)
1	Trivial	•Existing code, designed for reuse	
3	Low	•Existing code, <u>not</u> designed for reuse •Minor detailed design changes	
6	Medium	•New code required •Minor architectural change	
9	High	•Must reverse engineer code first •Significant architectural change	

Preferred but more complex, likely to meet cultural resistance

# Estimating Technique: Planning Poker

- Estimating the user stories for a release.
  - A release is one or more iterations.
- Entry Criteria:
  - Stories written and assigned to an iteration
  - Each release is identified and each has verification objectives;
  - Stories have been discussed with the team.
- Steps
  - Each estimator is given a deck of cards, each card has a valid number such as (1, 2, 3, 5, 8, 13, 21, ? or  $\infty$  = too big / not enough information)
  - The teams read the stories
  - An “typical” (average) story is selected
  - The story is read to the team and discussed briefly (a variation of the Wideband Delphi approach, <http://www.stellman-greene.com/aspm/content/view/23/38/>)
  - Each estimator selects a card to reveal his estimate
  - Cards are turned over so everyone can see them
  - Differences in estimates are discussed; especially outliers
  - Re-estimate until estimates converge



# Exercise #1: Estimate This!

Backlog Item	Relative Estimate
Create a 50 slide presentation on agile practices	
Read a James Patterson novel (500 pages)	
Read a bedtime story to a child	
Write a 6-8 page article on your latest software project and lessons learned	



Time: 20 minutes

# Check Point

- What we've covered so far
- Questions
- How are we doing?



Time: 5 minutes

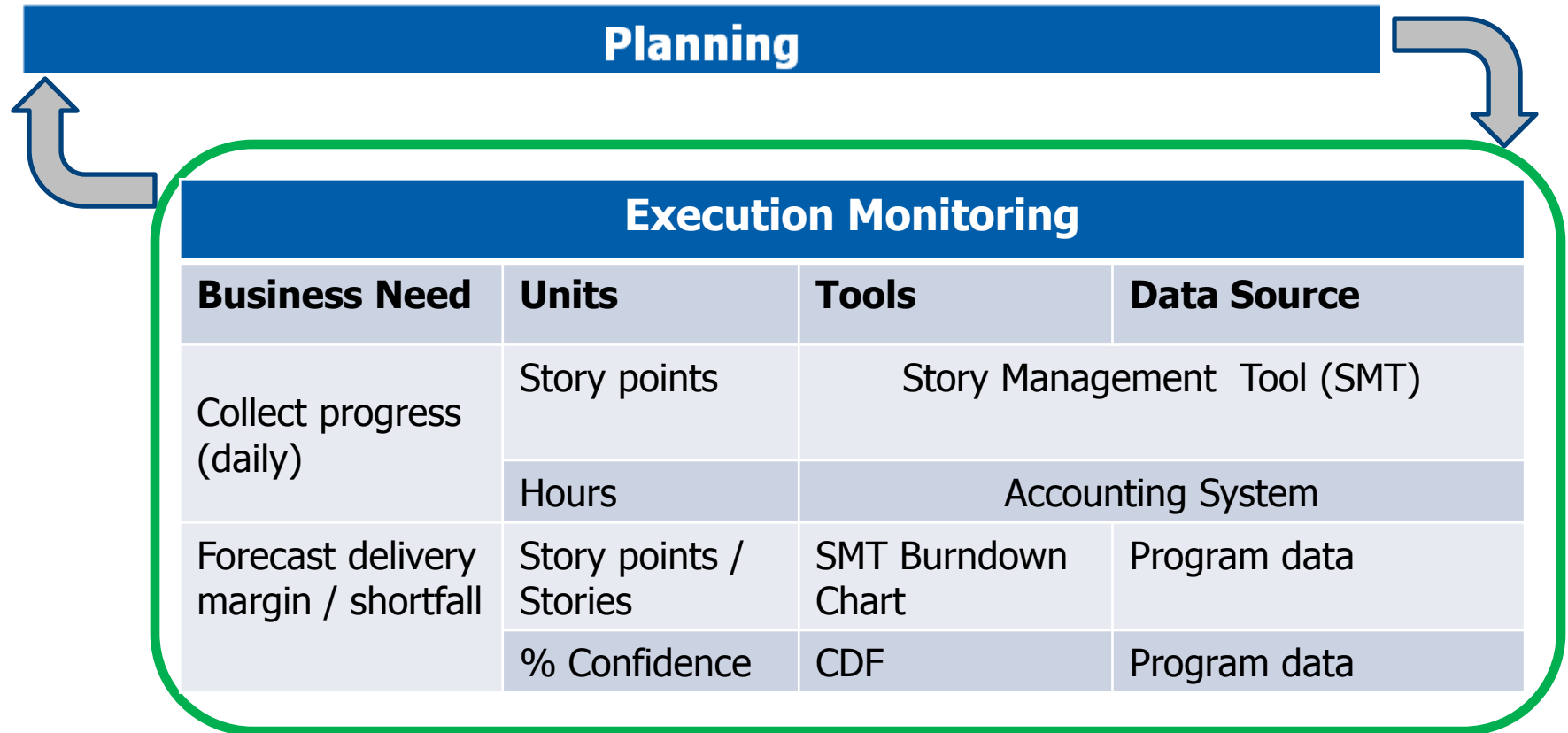
# Agenda

- Introductions
- Agile life cycle overview
  - Establish a common vocabulary, metrics overview, break
- Planning measures
  - Stories, estimation, exercise
- **Execution metrics & monitoring**
  - Burn down, velocity, exercise, break
- Retrospectives
  - Impounds, predicting next iteration
- ICM Review / Update Recommendations
- Closing comments



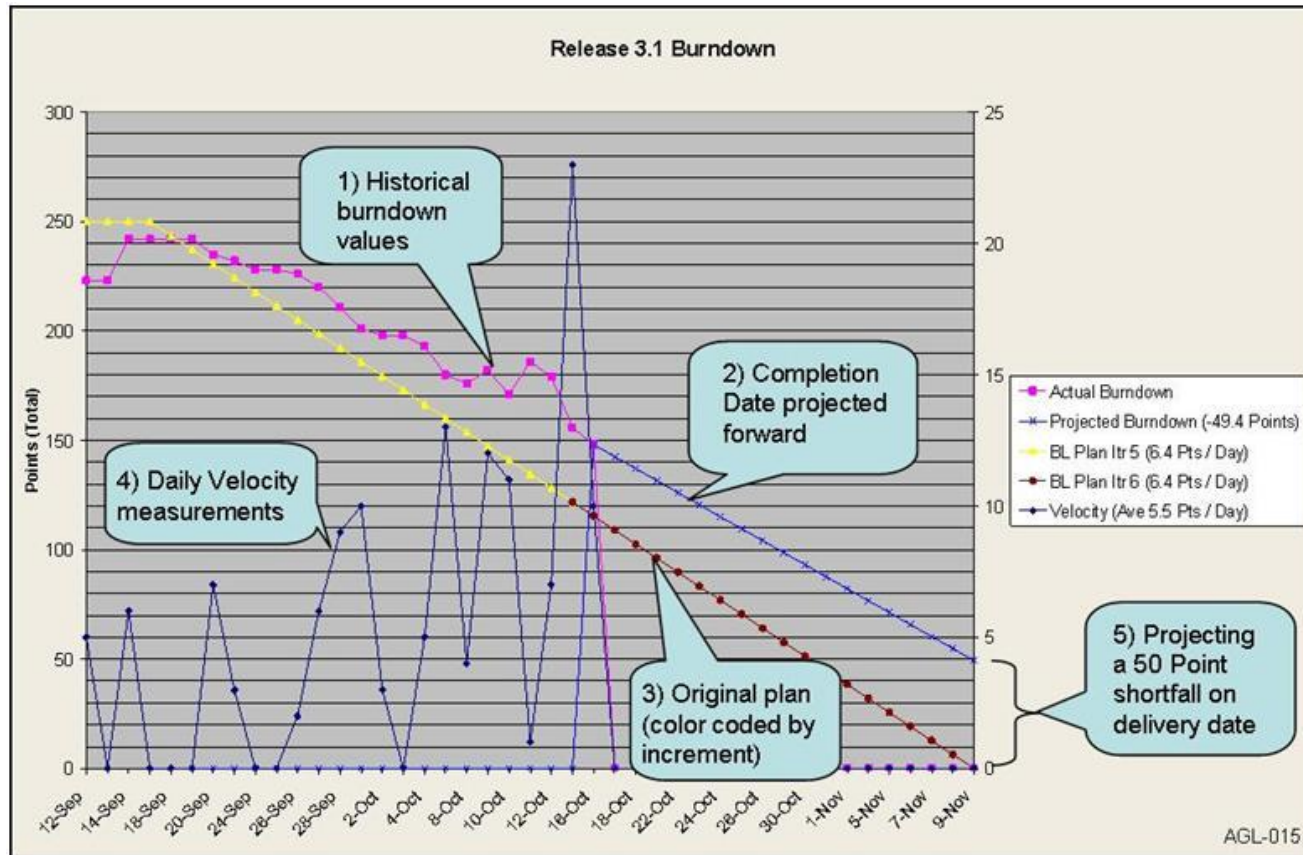
Goal: Introduction to agile SW development & metrics

# Metrics – Planning to Execution relationship



Next: Moving to Execution Metrics

# Execution Metrics & Monitoring - Burndown



Typical burndown chart (Spreadsheet tool\*)

1. Daily burndown
2. Projected completion. Average, nothing fancier
3. Original Plan by iteration
4. Daily Velocity
5. Projected shortfall prompts interesting discussions.

Other considerations:

1. Recommend **daily** velocity & hours collection.
2. ESLOC collection useful for comparison to non-agile programs.

\* <http://www.userstories.com/products>

Burndown metrics provide rapid status for everyone (all shareholders)

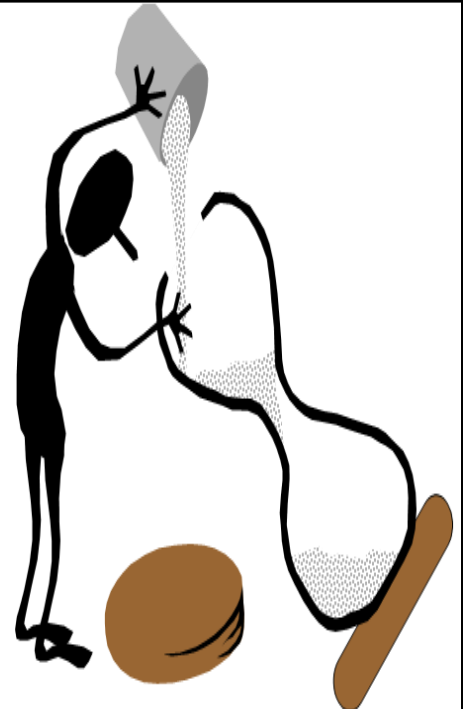


# Velocity – Operational Definition

Name (ID)	Size - Velocity Extension
Purpose	Velocity is the key performance measure to indicate progress toward capability delivery. Projected end date is driven by the project's experience with velocity.
Description	Velocity is the change in story points worked off over time. The average velocity (arithmetic mean) is used to forecast the delivery date based on the iteration's performance to date. Projected end date gives a current estimate of when the iteration will complete, along with story point over/under capacity based on project experience.
Base measures	<p>Periodicity of these measures is set by the project but, in general, should be measured daily. Extreme projects may use on-demand, instantaneous measurement systems.</p> <ol style="list-style-type: none"><li>1. EST = Total estimate. The overall scope, in story points, to be done in a given timeframe. This can go up and down as capacity (points) are added to the delivery.</li><li>2. ETC = ETC. The remaining amount of work to be done. The range of the values is from zero to EST.</li></ol>
Computation method	<ol style="list-style-type: none"><li>1. <math>\text{Velocity} = \text{ETC (yesterday)} - \text{ETC (today)} + \text{Story Point Added/Removed}</math></li><li>2. <math>\text{Average Daily Velocity} = \text{Sum of daily velocity} / \text{number of days worked}</math></li><li>3. <math>\text{Projected End Date} = \text{Today} + [\text{ETC (today)} / \text{Average Daily Velocity}]</math></li></ol>
Analysis Performed	<ol style="list-style-type: none"><li>1. Compare projected end date with program commitments and consider reducing or adding scope as appropriate.</li><li>2. Monitor actual burn down to planned burn down, modulating staff or scope as program needs dictate.</li></ol>

An agile unique metric

# Forecast Delivery margin / shortfall example

Agile Method	Measureable items are "Points" directly correlated to work products. A simple set of reports reflect position and contingency (slack) of time now and the through the future forecast . These same reports are the basis for "What If" analysis.	
A Few Traditional Methods	Slack Process Shortfall	
Margin Task(s) in plan	Single collection point(s) within Schedule (high probability failure). How to determine Risk verses Rate of Consumption for each task (if baseline duration is 20 days, what remaining duration value represents a "Yellow" rating).	
Three Point TEs (Time Estimates)	$(a + (4*m) + b)) / 6$ , Few take the initiative to collect historical verses plan and determine true 3 points (to much time)	
Baseline the "Late Dates", work to the Early Dates	Words from previous experience "I am not late, until I hit the baseline date". Unfortunately the contingency time is not retained but consumed automatically.	
Extract a contingency percentage	Speaking an a Master Planner/ Scheduler, if the Lead(s) know this is your plan, they will add 10%. If not, once you extract the percentage, the Leads will argue that you "do not have an achievable plan" and "lose" their buy-in.	
Genuine Earned Value	Tracking the number and rate of hours consumed in relation to the work preformed is sound but the measured item is Hours verses dates, not Products Developed verses dates.	

Working with the trees in the forest, not each green leaf

In review of traditional methods, there is a level of difficulty in calculating the quality and quantity of the product to be delivered.



- The “Point” system allows measurements that reach across the effort without dropping down to the nitty-gritty level.
- Used in conjunction with a institutionalized status process (weekly), the forecast will reflect “Points At Completion” versus the “Baseline Point Set”
  - A positive value is “missing” the target
  - A negative values is “more” is being produces at the current rate
  - For each situations, trend analysis will contribute to forecast validation

Directly tied to work effort

# Another benefit of flexibility, Change Control

In review of the “Points” exercise:

- All Technical Owners (Engineering (all departments), Cost Account Managers, Business Office) have developed the “Points Value”
- Using “Points” (not hours), the monitoring method is “lifted” from EXCRUCIATING detail to a level that is quickly measured and evaluated
- If a change to the plan is required (can you say Customer Re-direction), by using the “Points” system, the program can develop an impact plan and capture “Horse Trading Options” much quicker than using hours.
- Example, Initial Plan, Iteration #2 contains 29 Points (3, 3, 5, 9, 9)
  - Prior to closing and preceding with the iteration, the customer request a change (that once sized) is 11 points
  - Options:
    - Hold current end date, trade 11 for 11 (3, 3, 5)
    - Come in a little early (or less risk) with a trade of 11 verses 12 (9, 3)
    - Or push out by adding 11+29, plotting 40 point on the scale and determine the new forecast date



Trees versus leaves

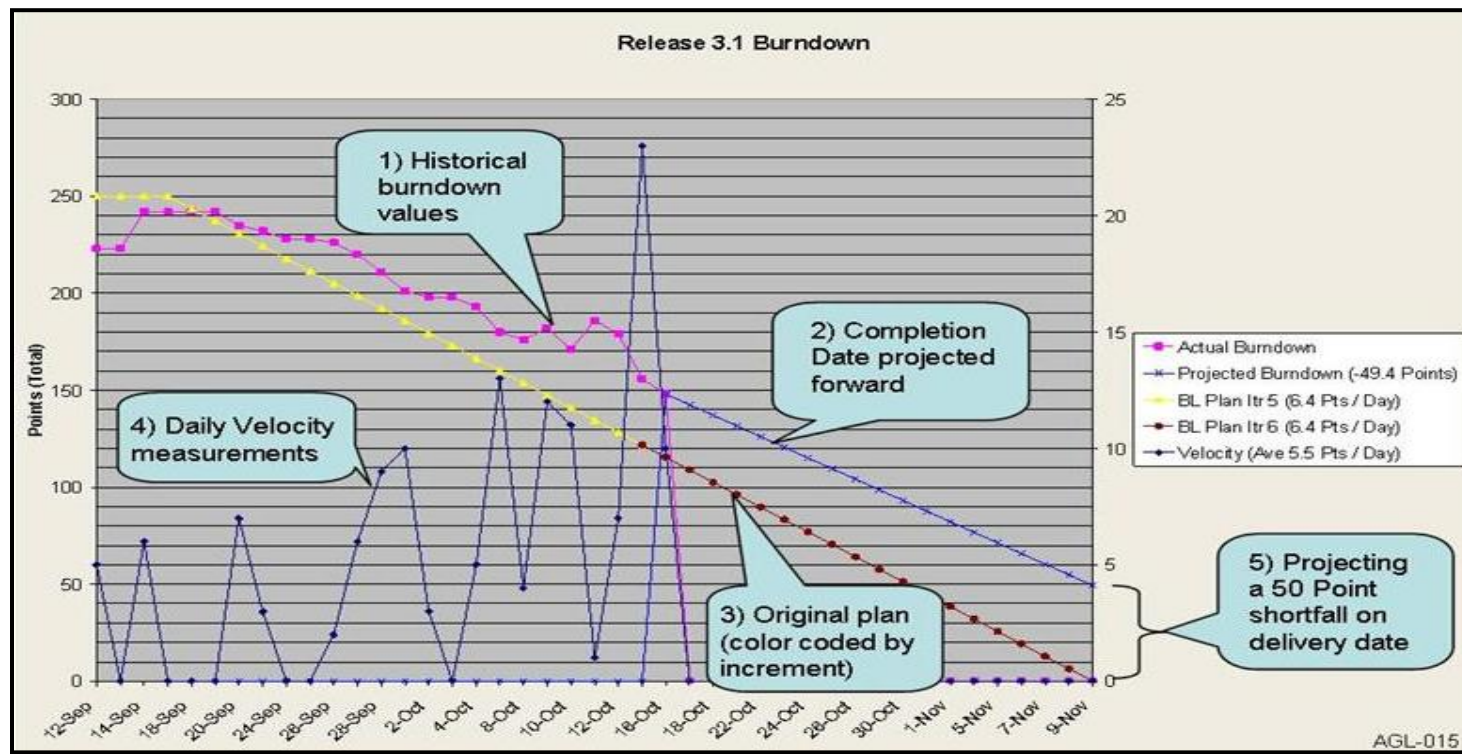
# Implementation of Measurement Method

## Example captures:

- The original plan “Points” value
- Rate and “speed” point are completed
- The forecast of “points complete” by target date

## This then is a source for any “change” evaluation

- The impact via to the points in the timeframe
- Change assessment, if already at risk (per example), sizing the INCREASED RISK if the change is implemented



Simplistic summary of “Time Now” and Forecast

## Exercise #2: Explain This!

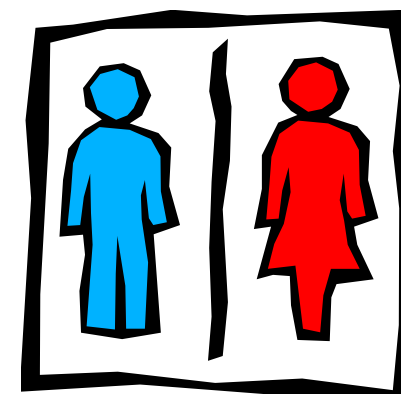
Exercise	Data values	Calculate these values:	Management, customer discussion points
Project: Video Game	Points to go 109 Ave daily 4.7 Days to go 21	Days needed _____ Points over / under _____ Days over / under _____	
Project: Online Dating	Points to go 225 Ave daily 6.3 Days to go 20	Days needed _____ Points over / under _____ Days over / under _____	

1. What do you advise management to do in each scenario?
2. What do you look for during retrospectives?
3. What velocity do you recommend for next iteration?

Time: 20 minutes for both scenarios



# Break



Time: 15 minutes



# Agenda

- Introductions
- Agile life cycle overview
  - Establish a common vocabulary, metrics overview, break
- Planning measures
  - Stories, estimation, exercise
- Execution metrics & monitoring
  - Burn down, velocity, exercise, break
- Retrospectives
  - Impounds, predicting next iteration
- ICM Review / Update Recommendations
- Closing comments



Goal: Introduction to agile SW development & metrics



# End of Iteration Retrospective, Data Impound

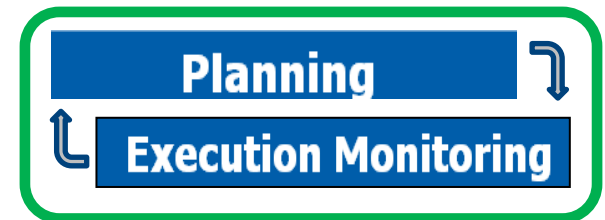
Iteration	Data Source
Proposal	Enterprise history comparable programs
1st Iteration	Team Estimates Enterprise history
Subsequent Iterations	Program historical Program historical + Enterprise history



Metrics staff impounds data now

# Additional End of Iteration Metrics of Interest

- ESLOC
  - A common “size” base measure to compare to traditional programs
- Productivity
  - Daily variation important in generating CDF for next iteration
  - End of iteration retrospective
- Velocity
  - The next Iteration can use the delivered velocity
  - No partial credit for incomplete stories
- Quality
  - Assess SW defects and leakage to end user



These metrics support next planning cycle

# How fast can we go?

- Process is defined
- Process performance is measured
- Can we deliver the *team* project in 59 days?
  - Customer wants 306 story points
    - Established via Planning Poker process.
  - Staff of 13.5 FTEs.
  - New, complex COTS integration.



***Next: use our tools and data to develop our confidence level***

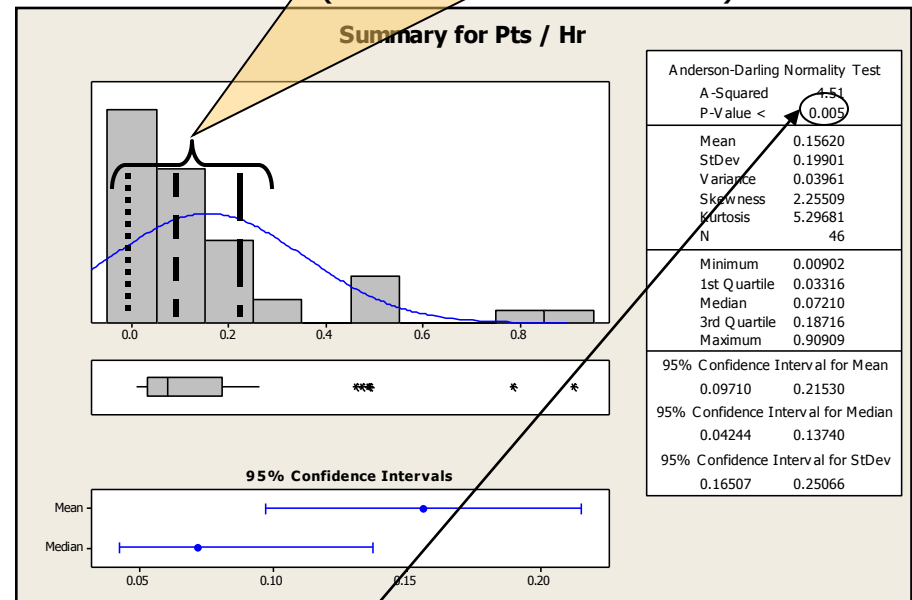
# Data Driven Decision Making

*Can team deliver 306 story points in 59 days?*

- Goal: Determine the Points per hour we can deliver at 3 favorite confidence levels:
  - 20% (Low probability of delivering on time)
  - 50% (Equal probability of delivering on early or late)
  - 80% (High probability of delivering on time)
- Step I. Quantify the Points / Hour we have demonstrated at the 3 levels above.
  - Then, given the hours we have (staff size \* days) we can determine the points we can produce.
  - The distribution is shown to the right
    - Minitab Command: Stat->Basic Statistics->Graphical Summary...  
Variable: Points / Hour

We want to be in between these amounts. The PM likes an 80% position.

Find Pts / Hr values we execute 80%, 50%, and 20% of the time  
(Note: Positions notional)

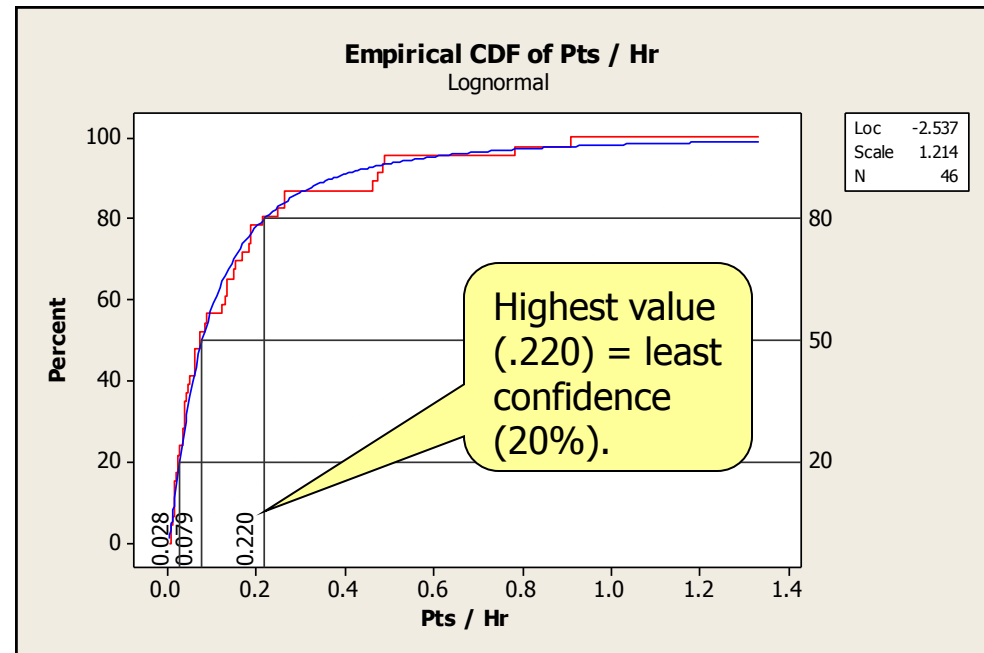


***Points / Hour data is not normally distributed. Hunch is lognormal***

# Data Driven Decision Making

*Can team deliver 306 story points in 59 days?*

- Step II: Develop a Empirical Cumulative Distribution Function to read the Points / Hour we have been experiencing.
  - Minitab Command: Graph->Empirical CDF... . Single. Variable: Points / Hours. Distribution: Lognormal. Scale->Percentile Lines->At Y Values: 20, 50, 80.
  - See results at right...
- Step III (Excel). Complete the algebra to determine the points that can be done at the various confidence levels.



Staff	Work Days	Hrs / Day	Total Hours
13.5	59	8	6372
Confidence	Points / Hr (Log Graph ->)	Points Done	
20%	0.22	1401.8	
50%	0.079	503.4	
80%	0.028	178.4	

306 points is between 50% and 80%

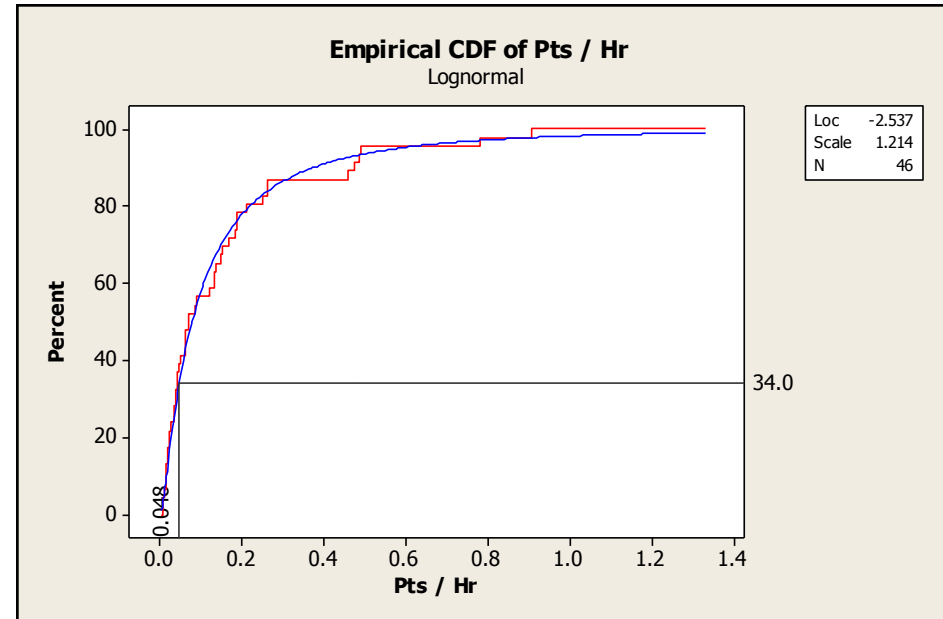
***We are between 50% and 80% confident, but how much?***

# Data Driven Decision Making

*Can team deliver 306 story points in 59 days?*

- Step IV: (Excel) Determine the Points / Hour needed to deliver 306 (0.0480).
- Step V (Minitab). Use the previous Empirical CDF to determine the % Confidence.
  - Minitab Command: Graph->Empirical CDF... . Single. Variable: Points / Hours. Distribution: Lognormal. Scale->Percentile Lines->At data Values: 0.0480
  - See result at right.
    - Final value =  $100 - 34\% = \mathbf{66\%}$

Staff	Work Days	Hrs / Day	Total Hours
13.5	59	8	6372
Confidence	Points / Hr (Log Graph ->)	Points Done	
20%	0.22	1401.8	
50%	0.079	503.4	
80%	0.028	178.4	
66%	0.048022599	306	Goal!



***We are 66% confident of delivering 306 story points in 59 days.***

# Management is now armed with data

- Step VI (Management) Risk Mitigation
  - Management reassured, but taking action:
  - Re-planning for a minimal delivery of 178 story points
    - Found an easy way to get to 220 story points as an initial delivery,
    - Still looking for a way to get down to 178...
    - Bringing on additional staff for Documentation work
      - Easily segregated, independent work.
  - Re-plan the balance of story points as our stretch goal.
  - Presented reduced plan to customer
    - Rational for change in base, stretch deliveries
  - RESULTS: Completed early, added additional capabilities back into delivery.



***Quantitative estimates guide delivery commitments***

# Agenda

- Introductions
- Agile life cycle overview
  - Establish a common vocabulary, metrics overview, break
- Planning measures
  - Stories, estimation, exercise
- Execution metrics & monitoring
  - Burn down, velocity, exercise, break
- Retrospectives
  - Impounds, predicting next iteration
- ICM Review / Update Recommendations
- Closing comments



Goal: Introduction to agile SW development & metrics



# Proposed ICM Changes (Page 1)

Information Category-Measurable Concept-Prospective Measures						
	Information Categories	Measurable Concepts	Questions Addressed	Measures		Notes
				Prospective Indicators	Sample Base Measures	
Project	Schedule and Progress	Milestone Completion	Is the project or service meeting scheduled milestones? Are critical tasks or delivery dates slipping?	- Milestone Progress	- Number of milestones started and completed versus plan	- Completion should be based on achieving specific quantifiable milestone completion criteria. - Milestones may include inch stones, or major critical milestones. - Might also look at critical path performance (slack time).
		Work Unit Progress	Are specific activities and products completed as scheduled?	- Requirements Progress - Problem Reports Progress - Reviews Progress - Change Requests Progress - System Elements or Units Progress - Test Cases Progress - Action Items Progress	- Requirements defined, traced, verified, validated - Problem reports discovered, closed - Reviews completed - Change requests opened, resolved - System elements or units designed, implemented, integrated, approved, qualified, accepted - Test cases developed, attempted, passed - Action items opened, completed	- Other work unit progress measures may be defined based on the work in progress. - Other schedule performance indicators are included with financial performance indicators (e.g. earned value measures).
		Work Backlog	Is the backlog of work units growing? Has the backlog of work units been adequately addressed?	- Work Unit Backlog Trends - Burndown Rates	- Work units in backlog, work units in backlog resolved	- Measure/categorize by priority level and age. - Work units may be: -- actions, assignments -- service requests -- story points or features -- maintenance actions -- open defects or open stakeholder problem reports
		Incremental Capability	Is capability being delivered as scheduled in incremental builds, releases, or service provisions?	- System Elements Integrated - Functionality Integrated	- Systems elements integrated (planned versus actual) - Functions integrated (planned versus actual)	
		Financial Performance	Is the project or service meeting budget and schedule objectives?	- CPI, SPI Trends - Earned Value Cost and Schedule Variance - Budget Adequacy and Trends - Cost Trends	- Earned Value: -- Budgeted Cost of Work Scheduled (BCWS) -- Budgeted Cost of Work Performed (BCWP) -- Actual Cost of Work Performed (ACWP) -- Budget at Completion (BAC) -- Latest Revised Estimate (LRE) -- Estimate at Completion (EAC) - Budget, planned, and actual costs	For deployed systems, costs include those to operate, maintain (resolve problems), and enhance system.
Resources and Cost		Personnel Effort	Is effort being expended according to plan? Is there enough staff with the required skills?	- Staff Level Sufficiency - Effort Distribution and Trends - Skill Profiles - Staff Turnover Rates	- Number of staff on project and projected - Number of staff by skill level - Number of staff by activity - Staff added, removed, quit	- Can also focus on key staff. - Effort distribution and trends by activity provides a more detailed profile. - Look at these measures for the current state and future projection. - Skills includes expertise, experience, training, education, and domain knowledge.
		Facilities and Support Resources	Are needed facilities, equipment, tools, and materials available as needed to meet milestones?	- Resource availability - Resource utilization	- Quantity needed, available - Time required, available, used	
		Technical Risk	Is the technical risk exposure at an acceptable level? Are the risk treatment actions performed per plan and are they effective?	- Risk Status - Risk Exposure Trends - Risk Treatment Trends	- Number risks by status and severity - Number risk treatment actions by status (new, in progress, closed) - Risk probability, impact, and criticality (to calculate exposure)	- Risk Treatment is also referred to as Risk Handling. - Technical impacts of risks that are realized could also be quantified (e.g. performance impacts). - Opportunities can also be identified and tracked, as well as enablers for those opportunities to occur.
Risk		Cost and Schedule Risk	Is the project realistic within established cost and schedule parameters? Is the project at risk of exceeding acceptable cost and schedule objectives?	- Schedule - Cost Impact Risk Trends	- Schedule risk	- Include updates as schedules and funding changes. - Develop a range of resource/Cost values with associated probabilities, not just one value. This range of potential - Cost impacts on schedule measured. - Could use the likelihood of exceeding specified percentage levels for cost and a specified number of months for schedule over the project baseline.

Agile addition in story points

Team capacity goes here (CDF)

Add delivery confidence (margin / shortfall)

# Proposed ICM Changes (Page 2)

Information Categories	Measurable Concepts	Questions Addressed	Measures		Notes
			Prospective Indicators	Sample Base Measures	
Size and Stability	Physical Size and Stability	How big is and how much change is occurring with the product's physical size, physical characteristics, or interfaces?	<ul style="list-style-type: none"> <li>- System Element Trends</li> <li>- Interface Complexity</li> <li>- Interface Compatibility</li> <li>- Lines of Code Trends</li> </ul>	<ul style="list-style-type: none"> <li>- System elements added, modified, deleted</li> <li>- Interface number (unique), complexity, growth, approval rates, changes, TBD/TBR closure per plan</li> <li>- Lines of code added, modified, deleted</li> </ul>	<ul style="list-style-type: none"> <li>- Consider both internal and external interfaces.</li> <li>- System elements can include software or hardware elements.</li> </ul>
	Functional Size and Stability	How much functional change is occurring with the product's functional characteristics?	<ul style="list-style-type: none"> <li>- Architecture Element Trends</li> <li>- Work Unit Backlog Size Trends</li> <li>- Function Points Trends</li> <li>- Call Center Request Trends</li> <li>- TBD/TBRs Trends</li> </ul>	<ul style="list-style-type: none"> <li>- Number added, modified, deleted</li> </ul>	<ul style="list-style-type: none"> <li>- This can be applied at any part or level of the system definition.</li> <li>- Functional architecture changes can be at the level of architecture description, model, or elements.</li> <li>- Call center requests can be categorized as problems</li> </ul>
Product Quality	Functional Correctness	Is the product good enough for delivery to the user? Are identified problems being resolved?	<ul style="list-style-type: none"> <li>- Defect Profiles</li> <li>- Defect Density</li> <li>- Technical Measurement Trends</li> <li>- System Elements Accepted</li> </ul>	<ul style="list-style-type: none"> <li>- Defects by status, severity, priority, distribution, age, etc.</li> <li>- Technical measurement requirement, target, threshold, budget, and actual</li> <li>- System elements verified</li> </ul>	Technical measurement includes Measures of Effectiveness, Measures of Performance, and Technical Performance Measures.
	Supportability - Maintainability	How much support does the system require? How difficult is it to support?	<ul style="list-style-type: none"> <li>- Time to Restore</li> <li>- Mean-Time-to-Repair</li> <li>- Cyclomatic Complexity</li> <li>- Utilization</li> </ul>	<ul style="list-style-type: none"> <li>- Hours to restore</li> <li>- Calendar hours and labor hours to repair</li> <li>- Number of paths through system</li> <li>- System element capacity available, used</li> <li>- Time for function (budget, actual)</li> </ul>	Support includes maintenance, training, provision of supplies, etc.
	Efficiency	Does the target system use a minimum of system resources?	<ul style="list-style-type: none"> <li>- Response Time</li> </ul>	<ul style="list-style-type: none"> <li>- System element capacity available, used</li> <li>- Time for function (budget, actual)</li> </ul>	It is important to capture benchmark times for key system functions. These can be reviewed as the system is maintained or altered, to ensure that no degradation occurs.
	Portability	To what extent can the functionality be re-hosted on different platforms?	<ul style="list-style-type: none"> <li>- Interface Compliance</li> </ul>	<ul style="list-style-type: none"> <li>- Interfaces verified</li> </ul>	Use of portability can also include reusability and
	Usability	Is the user interface adequate and appropriate for operations? Are operator errors within acceptable bounds?	<ul style="list-style-type: none"> <li>- User Interface Acceptability</li> <li>- Operator Error Trends</li> </ul>	<ul style="list-style-type: none"> <li>- Actions from user interface reviews</li> <li>- Operator errors</li> </ul>	
	Dependability - Reliability	How often is service to users interrupted? Are failure rates within acceptable bounds?	<ul style="list-style-type: none"> <li>- Mean-Time-to-Failure</li> <li>- Availability</li> </ul>	<ul style="list-style-type: none"> <li>- System element failures by severity, priority</li> <li>- System element start, end times</li> </ul>	Instead of availability, might measure downtime (outages).
	Security - Safety	How many vulnerabilities are identified and remediated by life-cycle phase? How many relevant attack patterns have been covered by test cases?	<ul style="list-style-type: none"> <li>- Profile of vulnerabilities</li> <li>- Cost to fix vulnerabilities</li> <li>- Attack Pattern Test Coverage Profile</li> </ul>	<ul style="list-style-type: none"> <li>- Vulnerabilities discovered, remediated</li> <li>- Cost to fix vulnerabilities</li> <li>- Test cases developed, verified per attack pattern</li> </ul>	
Process Performance	Process Compliance	How consistently does the project implement the defined project and enterprise processes?	<ul style="list-style-type: none"> <li>- Process Reference Maturity/Capability Rating</li> <li>- Process Audit Findings Distribution</li> </ul>	<ul style="list-style-type: none"> <li>- Maturity/Capability Rating Goal, Assessed</li> <li>- Number of audit findings by process area</li> </ul>	
	Process Efficiency	Are the processes efficient enough to meet current commitments and planned objectives?	<ul style="list-style-type: none"> <li>- Productivity Performance Trends</li> <li>- Cycle Time Performance Trends</li> <li>- Service Level Agreement (SLA) Response Trends</li> </ul>	<ul style="list-style-type: none"> <li>- Work unit size</li> <li>- Effort expended</li> <li>- Elapsed calendar and time expended</li> </ul>	For agile developments, team velocity is a measure of productivity.
	Process Effectiveness	Are the processes generating the results expected? How much rework is occurring?	<ul style="list-style-type: none"> <li>- Defect Containment</li> <li>- Test Effectiveness</li> <li>- Test Coverage</li> <li>- Defect-prone system elements distribution</li> <li>- Operational and Maintenance Effectiveness</li> <li>- Rework Effort Distribution</li> <li>- Rework System Elements Trends</li> </ul>	<ul style="list-style-type: none"> <li>- Defects by phase injected, discovered, and resolved (defect propagation)</li> <li>- Defects discovered per test case and test type</li> <li>- Defects discovered per system element</li> <li>- Schedule and effort expended - total and rework</li> <li>- System elements requiring rework</li> </ul>	<ul style="list-style-type: none"> <li>- Defects per system element is particularly important for key elements of the architecture, or if safety/security related.</li> <li>- For services, schedule and effort expended might include those related to service calls.</li> <li>- Rework in production might measure waste of production units</li> <li>- Could also measure benefits of processes (e.g. cost prevention)</li> <li>- Defect containment is also called "Defect Escapes"</li> </ul>
Technology Effectiveness	Technology Suitability	Can technology meet all allocated requirements, or will additional technology be needed?	<ul style="list-style-type: none"> <li>- Requirements Coverage</li> </ul>	<ul style="list-style-type: none"> <li>- Requirements met by technology</li> </ul>	
	Technology Maturity	Is the technology ready to be used in this project?	<ul style="list-style-type: none"> <li>- Technology Maturity Trends</li> </ul>	<ul style="list-style-type: none"> <li>- Technology readiness level (TRL)</li> </ul>	Might also consider technology obsolescence - is the technology about to become obsolete?
	Technology Volatility	Does new technology pose a risk due to too many changes?	<ul style="list-style-type: none"> <li>- Technology Baseline Changes Trends</li> </ul>	<ul style="list-style-type: none"> <li>- Number of requirements impacted by changed technology</li> </ul>	
Customer Satisfaction	Customer Feedback	How do our customers perceive our performance for individual projects and the enterprise? Are we meeting user expectations?	<ul style="list-style-type: none"> <li>- Satisfaction Ratings Trends</li> <li>- Award Fee Distributions</li> </ul>	<ul style="list-style-type: none"> <li>- Satisfaction ratings</li> <li>- Award fees received</li> </ul>	Contractor Performance Assessment or other survey
	Customer Support	How quickly are customer support requests being addressed?	<ul style="list-style-type: none"> <li>- Support Request Distributions</li> <li>- Support Time Trends</li> </ul>	<ul style="list-style-type: none"> <li>- Number of support requests</li> <li>- Calendar time to address requests</li> </ul>	

Add discovery metrics, size in story points

Add defects, peer reviews stratified by story points

# Proposed ICM Changes (Page 3)

	Information Categories	Measurable Concepts	Questions Addressed	Measures		Notes
				Prospective Indicators	Sample Base Measures	
Enterprise	Schedule & Progress	Milestone Completion	Are the projects or services within this enterprise on track? Which ones need management attention (which ones are most behind)?	- Enterprise Milestone Status	- Number of milestones started and completed versus plan (percent complete versus plan)	- Milestones mean major milestones or sum of set of milestones. - For the enterprise, want some early indication of whether major milestones will be met, for each project in the enterprise (or those strategically important or those most at risk).
		Work Backlog	What is the enterprise work backlog? What should be scheduled next?	- Work Unit Backlog Trends - Burndown Rates	- Work units in backlog by priority level for major item - Work units in backlog resolved	Work units may be: - open stakeholder requests - defects - enhancements, needs - tasks - new contracts, RFPs - deliveries
	Resource Cost	Financial Performance	Is the enterprise receiving and spending money as planned? Does the enterprise financial process support the needs of the enterprise? Is the enterprise meeting its goals and objectives?	- Funding Availability - Disbursement / Obligation Rate Trends - Earnings Progress - Investment Capital	- Budget, planned, and actual funds available - Disbursements, obligations - Sales, costs, incentive/award fees, taxes - Contribution to overhead - Invested capital, additional revenue	Is funding available as needed? Consider: - spread of money across the year for multiple projects - color of money and plus-ups for government projects - funding blocks, pull-backs - studies, management reserve, as well as development and maintenance projects
		Personnel Effort	Within the enterprise, are there sufficient qualified people to satisfy commitments?	- Staff Level Sufficiency - Effort Distribution and Trends - Workforce Skills Profiles - Workforce Age Profiles - Staff Turnover Rates	- Number of staff by skill level and project assigned to project - Number of staff by age - Staff added, removed, quit	
		Facilities/C31 and Support Resources	Are needed facilities, equipment, tools, and materials available, across the enterprise? Where should future investments occur?	- Resource availability - Resource utilization	- Quantity needed, available - Time required, available, used	For the aggregate set across all projects.
	Risk	Technical Risk	Is the technical risk exposure for the enterprise at an acceptable level? Do we have a balanced risk/reward portfolio?	- Portfolio Risk Status - Risk Tolerance	- Number risks by status and severity	Should also consider enterprise risks, in addition, assessing aggregate risks across projects.
		Cost and Schedule Risk	Is the enterprise at risk of exceeding acceptable cost and schedule objectives?	- Schedule Impact Risk Profile - Cost Impact Risk Profile	- Schedule Risk - Cost Risk	
	Size & Stability	Physical Size and Stability Functional Size and Stability	How many (unique) platforms, systems, or applications are in development, maintenance, operations? Are they compatible, where needed?	- Platform/System Trends - External and Cross-Platform Interface Complexity and Compatibility	- Number of unique platforms, systems, or applications - Interface number (unique), complexity, growth, changes	
	Product Quality	Functional Correctness Dependability-Reliability	Is the set of projects delivering quality products that meet user expectations? Are known problems being resolved?	- Stakeholder Defects Distribution - Stakeholder Requirements Validation Profile - Warranty Trends	- Defects by status, severity, priority, distribution, etc. - Number of stakeholder requirements Validated Successfully - Warranty claims	Stakeholder defects are those identified after fielding.
	Process Performance	Process Compliance	Are enterprise processes being applied across the enterprise?	- Reference Maturity/Capability Profile - Process Audit Findings Distribution - Exception Distributions	- Maturity/Capability Rating Goal, Assessed - Number of audit findings by process area - Number of exceptions by process element	Exceptions include waivers and amount of tailoring.
		Process Efficiency	What are enterprise norms for completing life-cycle activities (schedule, cost, performance)? Do the majority of projects meet the norms?	- Productivity Baselines and Trends - Cycle Time Baselines and Trends	- Work unit size - Effort expended - Elapsed calendar and time expended	
		Process Effectiveness	Are the enterprise processes sufficient to accomplish enterprise objectives? How much rework is occurring?	- Rework Effort Distribution - Rework System Elements Distribution	- Schedule and effort expended - total and Rework	Rework in production might measure waste of production units
Technology Effectiveness	Technology Maturity	Does the enterprise have sufficient technology management plans and implementations? Is technology investment in place to ensure adequate leverage of technology into projects?	- Technology investment versus plan - Needs Met by Technology Insertion - Technology Refresh Rate	- Investment amount - Number of needs met by inserted technology - Technologies replaced		
	Customer Satisfaction	Customer Feedback	How do our customers perceive the enterprise's set of products (product lines)? Are they meeting user expectations?	- Satisfaction Ratings Trends - Market Share - Value for Money (government)	- Satisfaction ratings - Enterprise sales, total market sales, new contracts awarded - Assessed value	- Generally measured through a survey. - Government is focused on mission accomplishment (versus market share or investment)

Add end of iteration impounds (capacity, defects, etc)

# Agenda

- Introductions
- Agile life cycle overview
  - Establish a common vocabulary, metrics overview, break
- Planning measures
  - Stories, estimation, exercise
- Execution metrics & monitoring
  - Burn down, velocity, exercise, break
- Retrospectives
  - Impounds, predicting next iteration
- ICM Review / Update Recommendations
- Closing comments



Goal: Introduction to agile SW development & metrics

***NORTHROP GRUMMAN***

The logo consists of the words "NORTHROP GRUMMAN" in a bold, blue, italicized sans-serif font. A solid blue horizontal line is positioned directly beneath the text.

Backup slides

# Stories – compared to other benchmarks

Technique	Technique Description
"Classic" Requirements Development	<p>The definition and refinement of system-, subsystem-, and lower-level functional and performance requirements and interfaces to design the system.</p> <p>Used with <b>Logical Analysis</b> and <b>Design Solution</b> technical processes. These processes iterate at each level of the system structure, and then applied recursively to lower levels of the physical architecture throughout development.</p>
Use Case	A <b>description of a system's behavior</b> as it responds to a request that originates from outside of that system.
Stories	A description of functionality that is meaningful and valuable to a <b>user of the system</b>

***Stories define the system from a business perspective***



# Requirements to User Stories

The system shall provide the capability for making hotel reservations.

As a premiere member, I want to search for available discounted rooms.

As vacationer, I want to search for available rooms.

As vacationer, I want to save my selections.

# Non-Functional Requirements?

As a vacationer and user of the hotel website, I want the system to be available 99.99% of the time...

As vacationer, I want web pages to download in <4 seconds...

As the hotel website owner, I want 10,000 concurrent users to be able to access the site at the same time with no impact to performance...

**Stories for  
non-functional  
requirements**

**Describes  
system  
behavior or  
characteristics**



- Often written by the Product Owner or as a team
- Brainstorm to generate ideas
- Some stories start out as epic stories; break them down
- Stories should be drafted and estimated prior to the release planning meeting

# References and Suggested Reading List

## *Creating Adaptive Businesses*

- Adaptive Enterprise – Steven Haeckel
- Complexity leadership theory: Shifting leadership from the industrial age to the knowledge era – Marion, R., McKelvey, B., & Uhl-Bien, M. (2007). *Leadership Quarterly*, 18(4), 298-318.

## *Agile Development Practices*

- Agile Project Management with Scrum – Ken Schwaber
- Agile Retrospectives – Esther Derby and Diana Larsen
- Agile Software Development: Adaptive systems principles and best practices – Meso and Jain
- Agile Software Development with Scrum – Ken Schwaber and Mike Beedle
- Agile Systems Development Lifecycle (<http://www.ambysoft.com/essays/agileLifecycle.html>) – Scott Ambler
- The Agile Journal. Agile Testing. <http://www.agilejournal.com/media-center/educational-videos/1188-borland-part1>
- XP Core Practices - Lindstrom & Jeffries (2004)
- Scrum and The Enterprise – Ken Schwaber
- [www.mountangoatsoftware.com](http://www.mountangoatsoftware.com) – Mike Cohn
- An introduction to agile and iterative project management - Hallett, D. (2006)

## *Managing Change*

- Fearless Change – Manns and Rising
- The Fifth Discipline – Peter Senge
- Leading Change – John Kotter

***NORTHROP GRUMMAN***

