Agile Metrics

Metrics

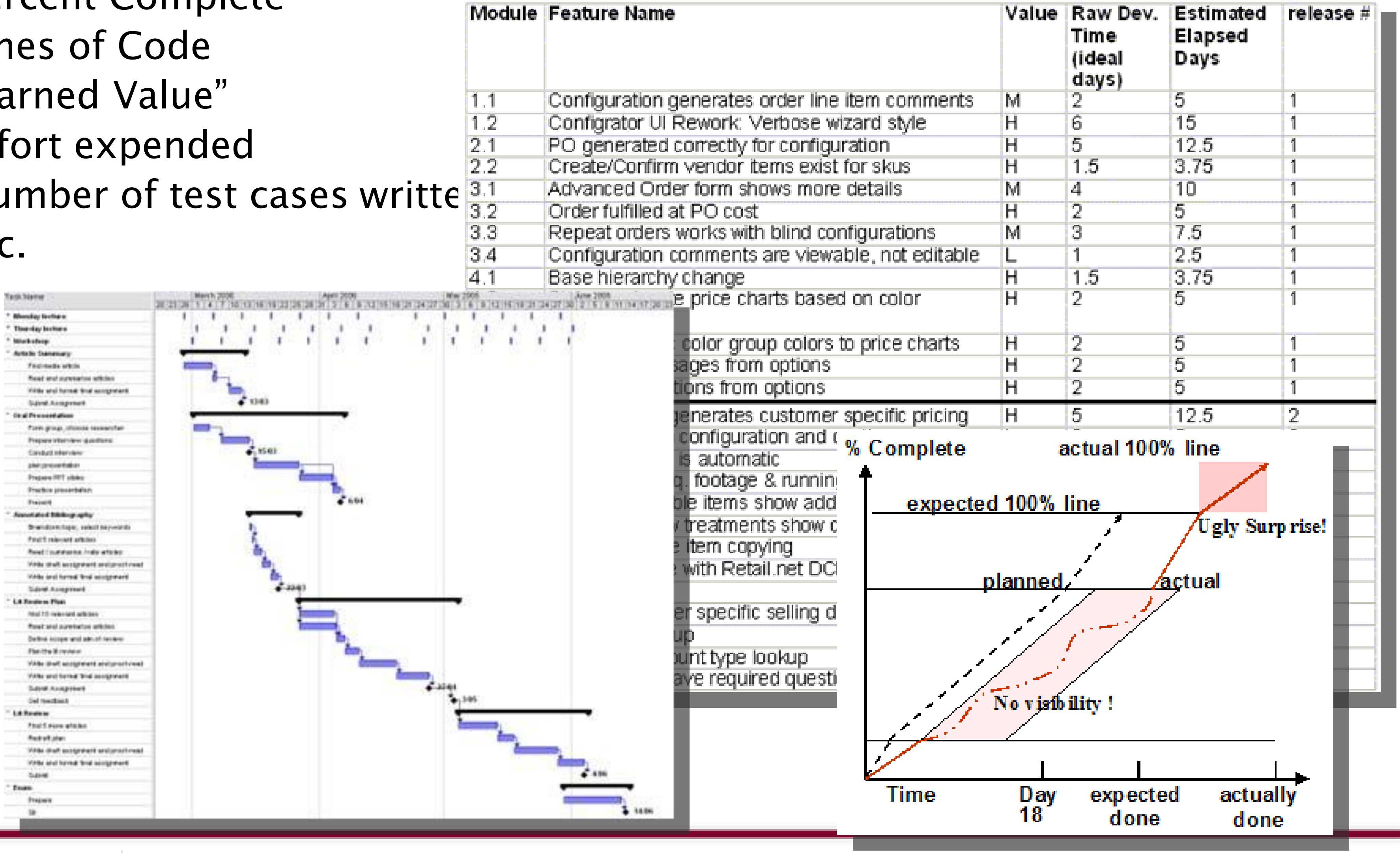
- Good metrics affirm & reinforce Agile principles
- Metrics are to measure outcome, not output
- Reporting should measure trends, not numbers
- ***Reports should provide fuel for meaningful conversations***
- This data should be easy to collect
- Good data is useful in gathering feedback on a more frequent basis
- Metrics should help define excellence vs. good enough production

© 2011 VersionOne

Traditional Metrics

Traditional Metrics typically look at:

- Percent Complete
- Lines of Code
- “Earned Value”
- Effort expended
- Number of test cases writte
- Etc.

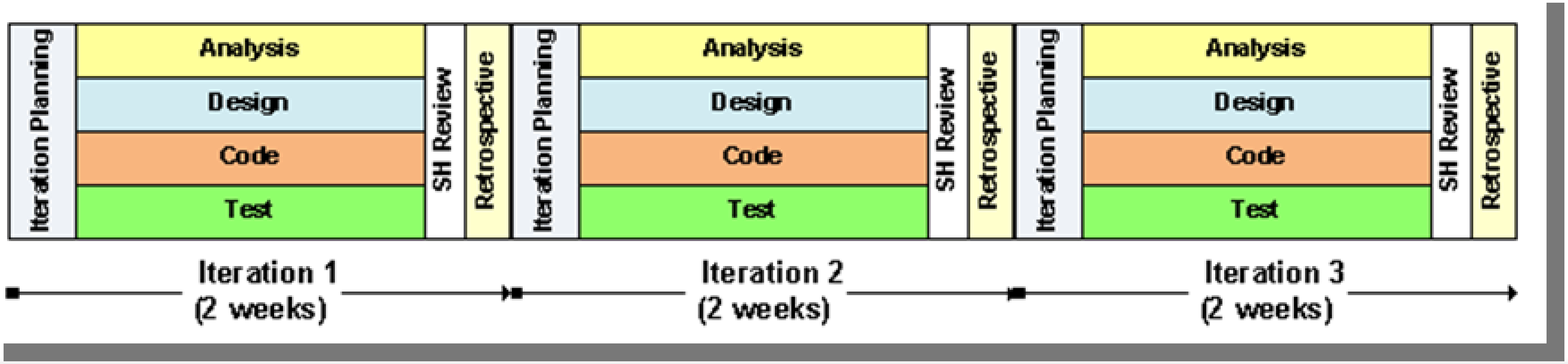


Module	Feature Name	Value	Raw Dev. Time (ideal days)	Estimated Elapsed Days	release #
1.1	Configuration generates order line item comments	M	2	5	1
1.2	Configrator UI Rework: Verbose wizard style	H	6	15	1
2.1	PO generated correctly for configuration	H	5	12.5	1
2.2	Create/Confirm vendor items exist for skus	H	1.5	3.75	1
3.1	Advanced Order form shows more details	M	4	10	1
3.2	Order fulfilled at PO cost	H	2	5	1
3.3	Repeat orders works with blind configurations	M	3	7.5	1
3.4	Configuration comments are viewable, not editable	L	1	2.5	1
4.1	Base hierarchy change	H	1.5	3.75	1
	e price charts based on color	H	2	5	1
	color group colors to price charts	H	2	5	1
	sages from options	H	2	5	1
	tions from options	H	2	5	1
	enerates customer specific pricing	H	5	12.5	2
	configuration and				
	is automatic				
	q. footage & runnin				
	ble items show add				
	treatments show				
	e item copying				
	with Retail.net DC				
	er specific selling d				
	up				
	unt type lookup				
	ave required questi				

© 2011 VersionOne

Why are metrics/reporting essential to Agile projects?

- A constant evaluation of progress is used to redirect priorities
- Delivering on commitment to stakeholders to keep them informed in a meaningful manner
- "Embracing Change" requires insight on those changes
- Automation is encouraged



© 2011 VersionOne

Agile Metrics differ from typical PM / SW Metrics

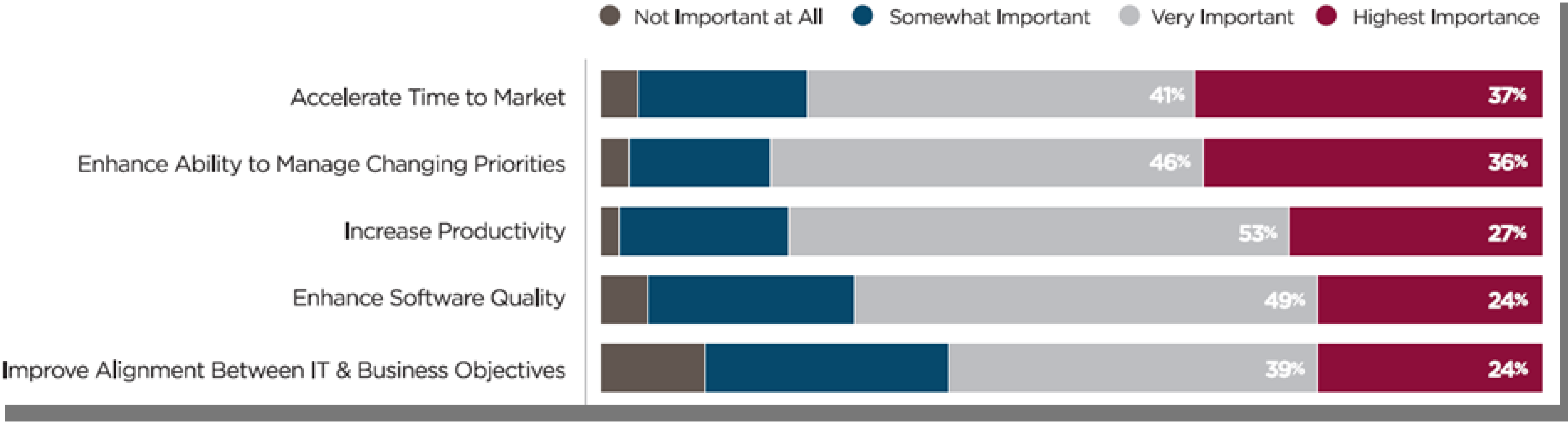
- Empirical – items are measurable.
- *'Done or not done' – not '62.3% complete'*
- Team understanding and acceptance of what is being measured
- Agile Metrics are used to deliver better software
- Agile metrics are not the end…they are the beginning of a discussion or a decision

You can still produce typical Project Management / Software Metrics, but they won't be as valuable.

© 2011 VersionOne

Why Go Agile?

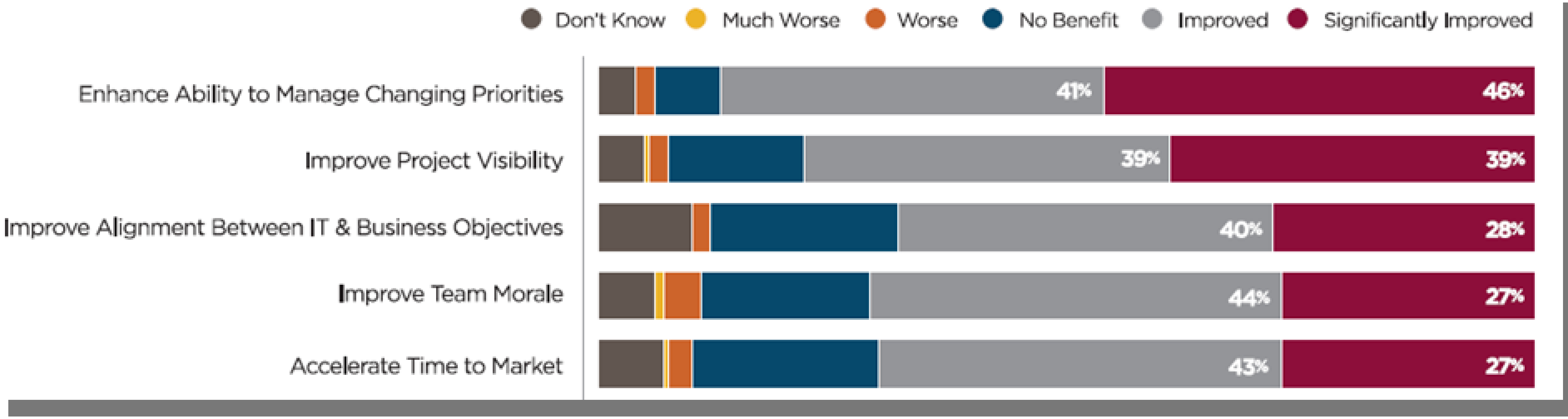
Survey's Top 5



© 2011 VersionOne

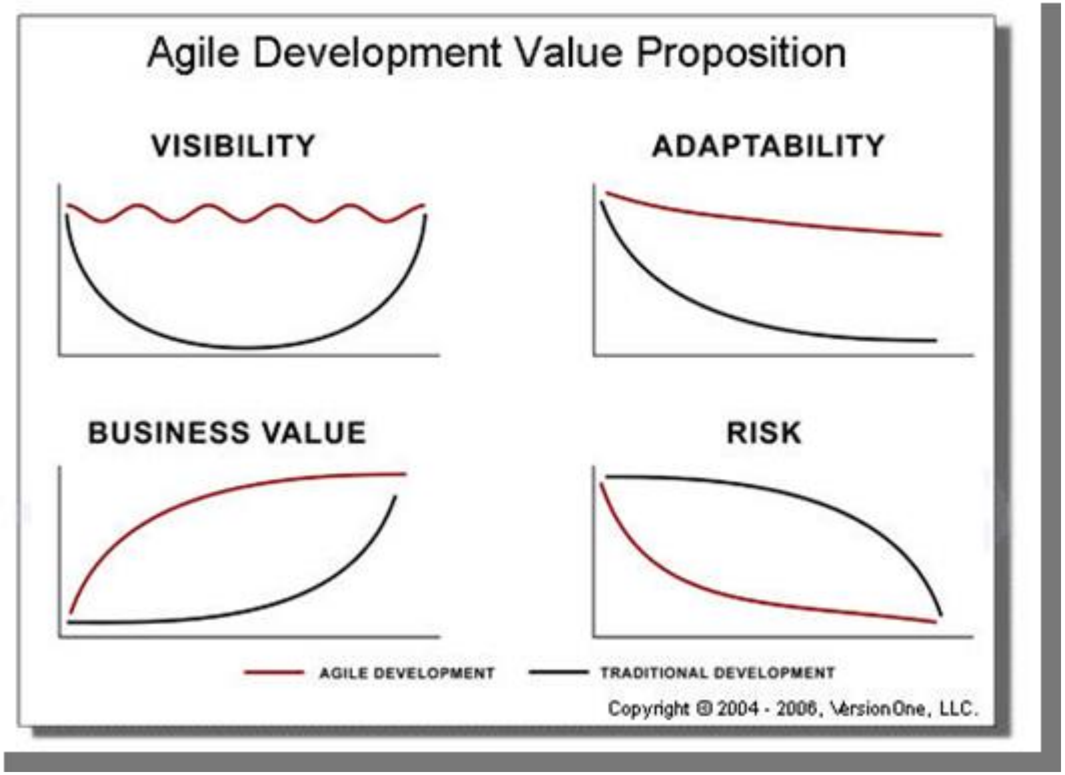
What Benefits are they Realizing?

Survey’s Top 5



© 2011 VersionOne

A Different Perspective



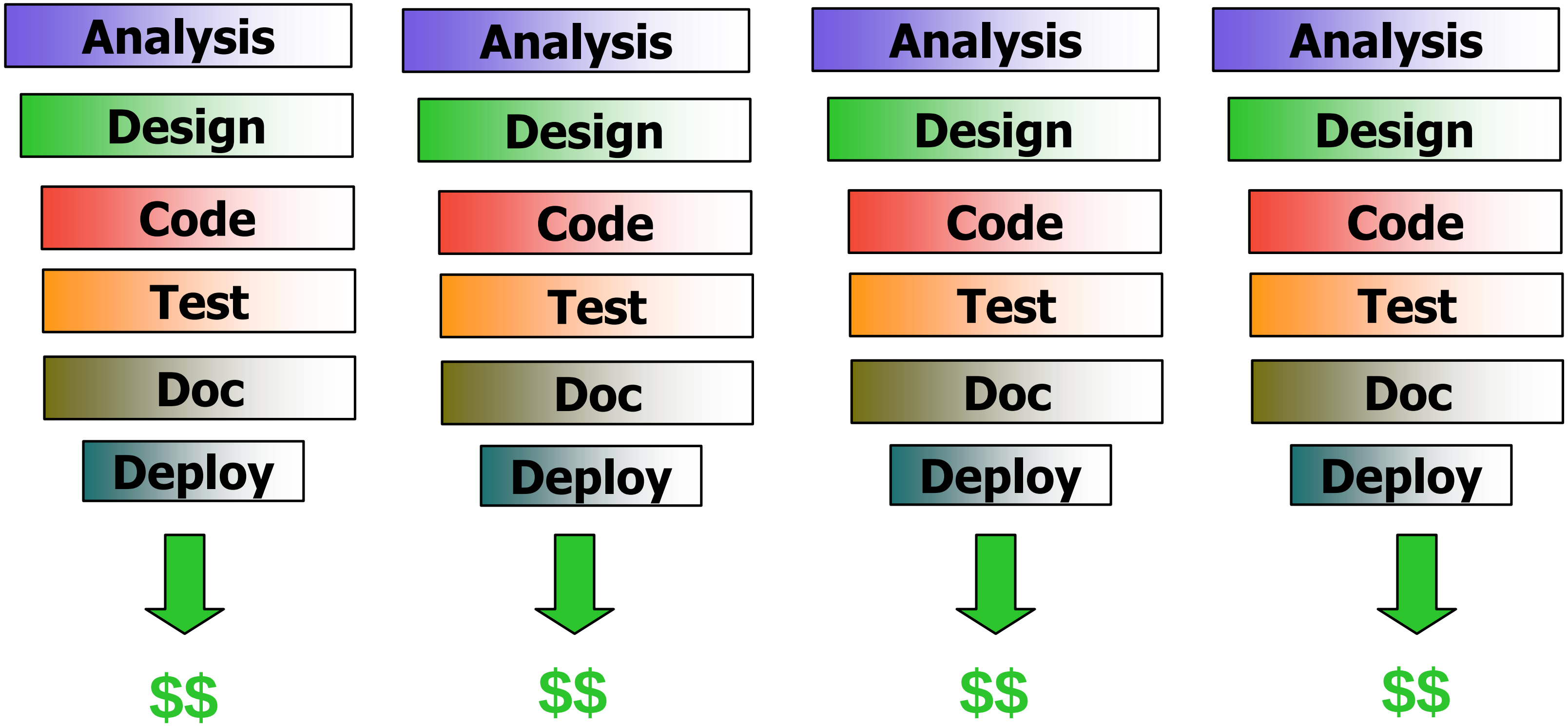
With plan-driven approach, we have a plan going into a project, but we quickly lose sight of what's going on

So we create surrogate measures that we hope are representative of the true health of the project.

Near the end of the project, the true state of the project emerges – were we right?

© 2011 VersionOne

Agile Projects Deliver Value Every Iteration/Release



© 2011 VersionOne

The Business Dilemma

Return on Investment

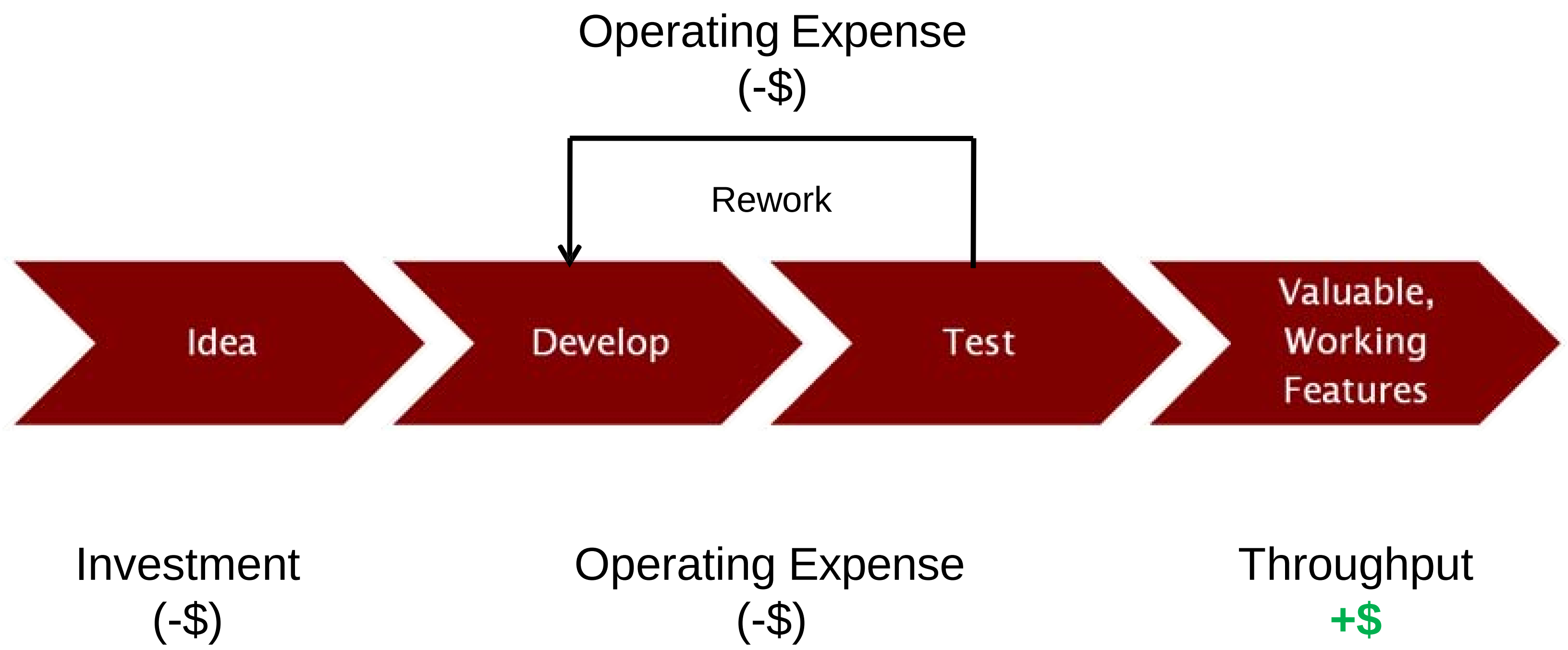
- What do we build?
- How do we maximize return while limiting the investment?

In order to have any Return,

- Something of **Value** must be produced, plus
- There must be an **Opportunity** to sell it

© 2011 VersionOne

Throughput Accounting: The Heart of the Business Case for Agile



Operating Expense
(-$)

Rework

Idea

Develop

Test

Valuable, Working Features

Investment
(-$)

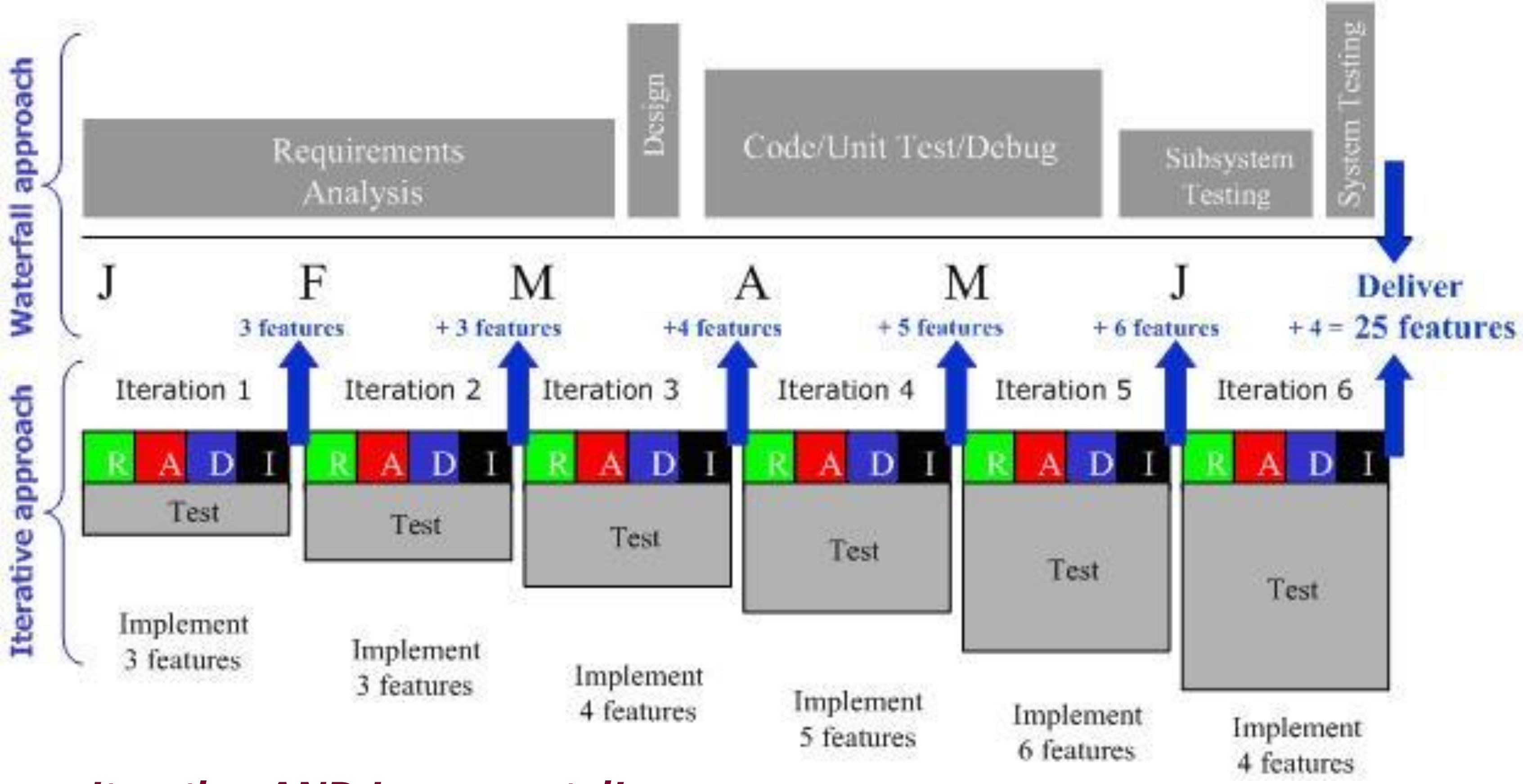
Operating Expense
(-$)

Throughput
+$

Maximize Throughput by removing system constraints while limiting Investment and Operating Expenses.

© 2011 VersionOne

Planning: A Comparison

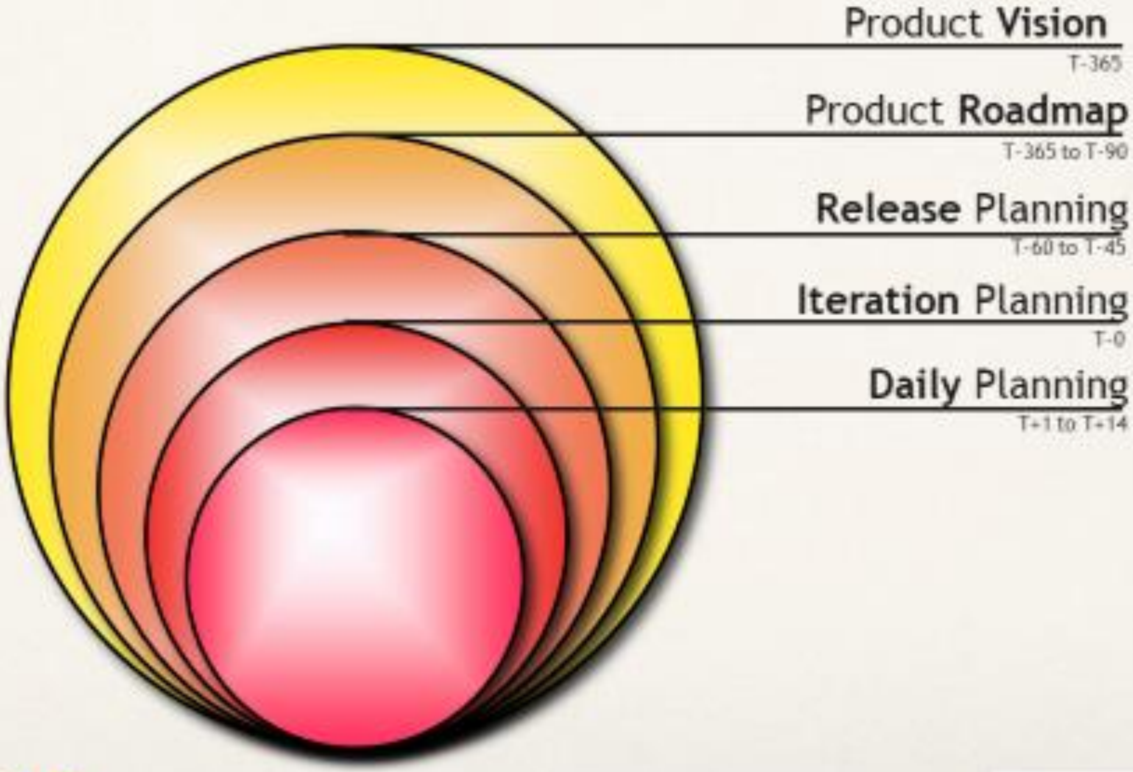


Iterative AND Incremental!

© 2011 VersionOne

The Five Levels of Agile Planning

Agile teams plan their projects at 5 levels:



Product **Vision**

T-365

Product **Roadmap**

T-365 to T-90

Release Planning

T-60 to T-45

Iteration Planning

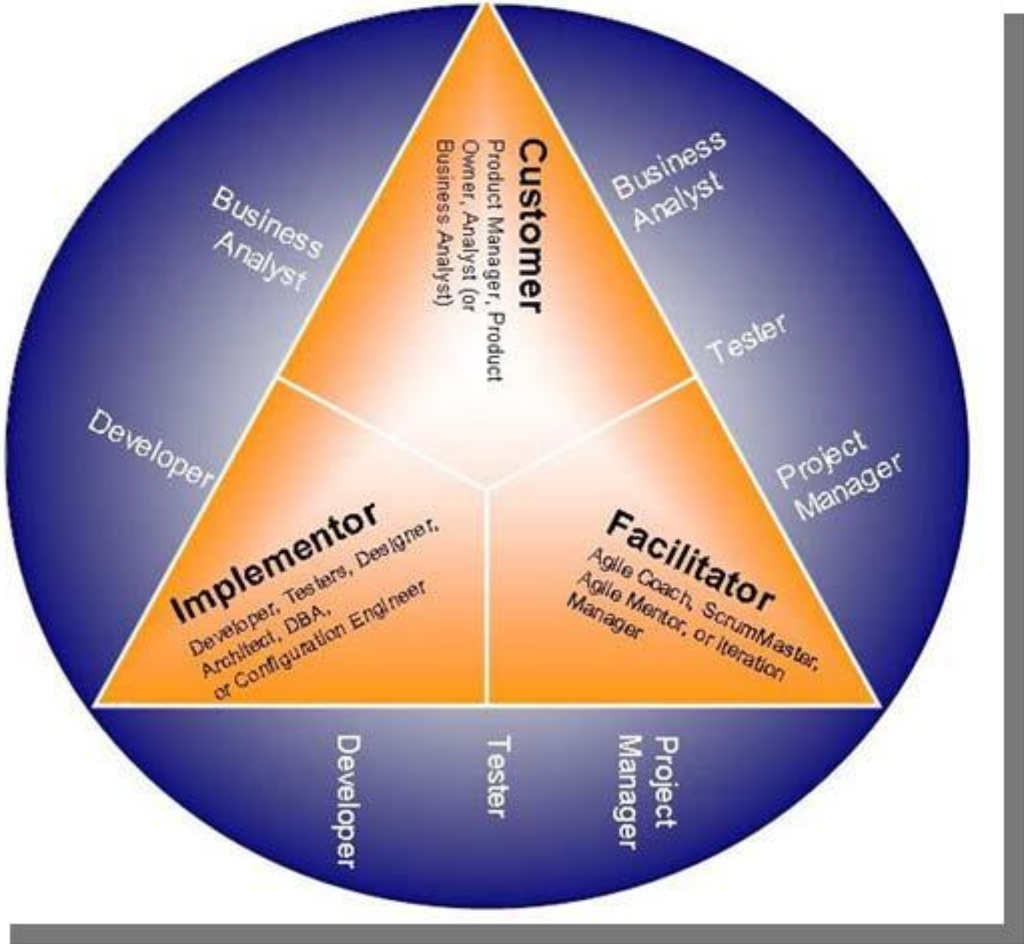
T-0

Daily Planning

T+1 to T+14

Copyright 2013 AgileDad LLC Licensed for Classroom Use Only.

Agile Roles in General


Customer
Product Manager, Product Owner, Analyst (or Business Analyst)
Business Analyst
Business Analyst
Tester
Developer
Project Manager
Implementor
Developer, Testers, Designer, Architect, DBA, or Configuration Engineer
Facilitator
Agile Coach, ScrumMaster, Agile Mentor, or Iteration Manager
Developer
Tester
Project Manager

http://kanemar.files.wordpress.com/

- Most Agile Methods profess the use of 3-5 different roles
- Many teams adopting Agile struggle to determine where their traditional role fits in an Agile landscape
- Every role fits into 3 Classes:
  - Customer
  - Facilitator
  - Implementer

Key metrics/reporting

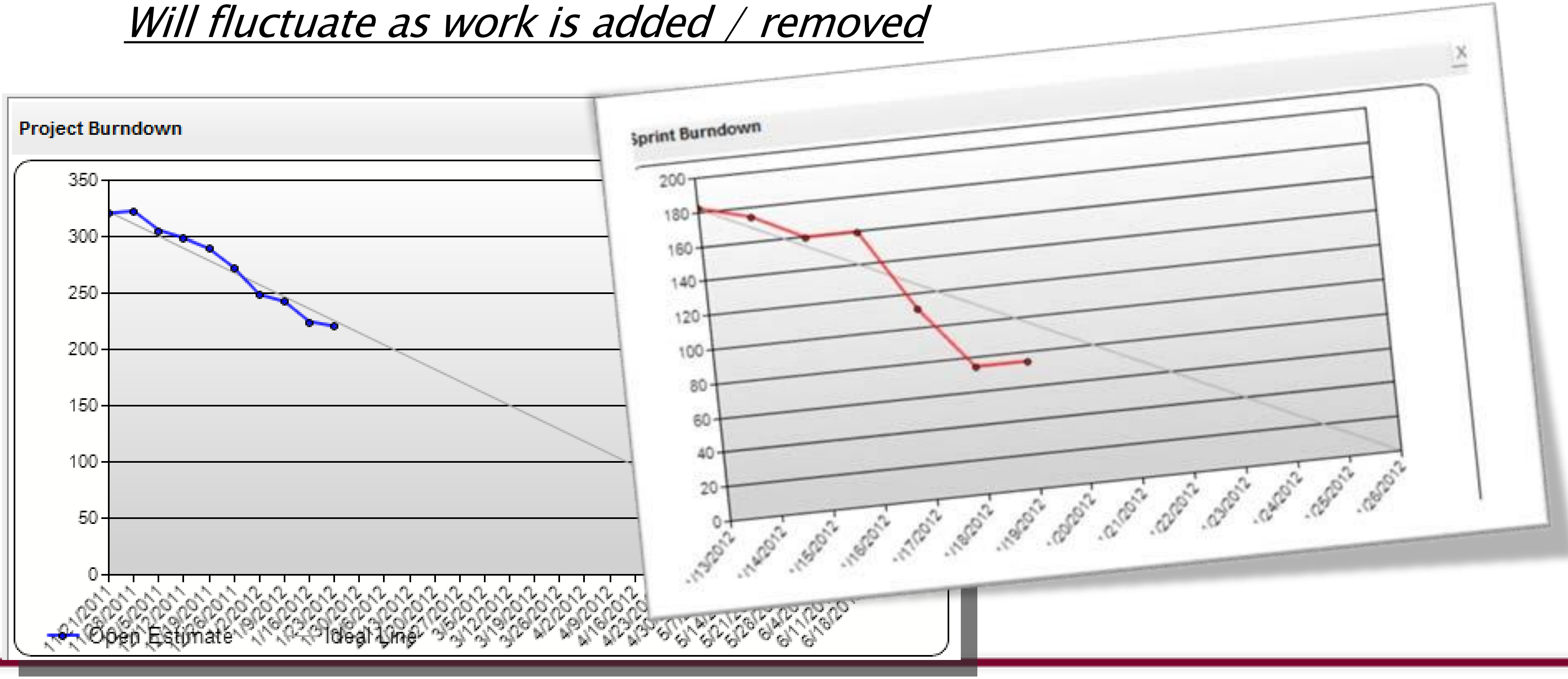
Some key Agile Metrics include:

- Burndowns
- Velocity Trend
- Counts and statuses of work items and defects
- Team Member Load, Effort
- Test Reports

Burndown

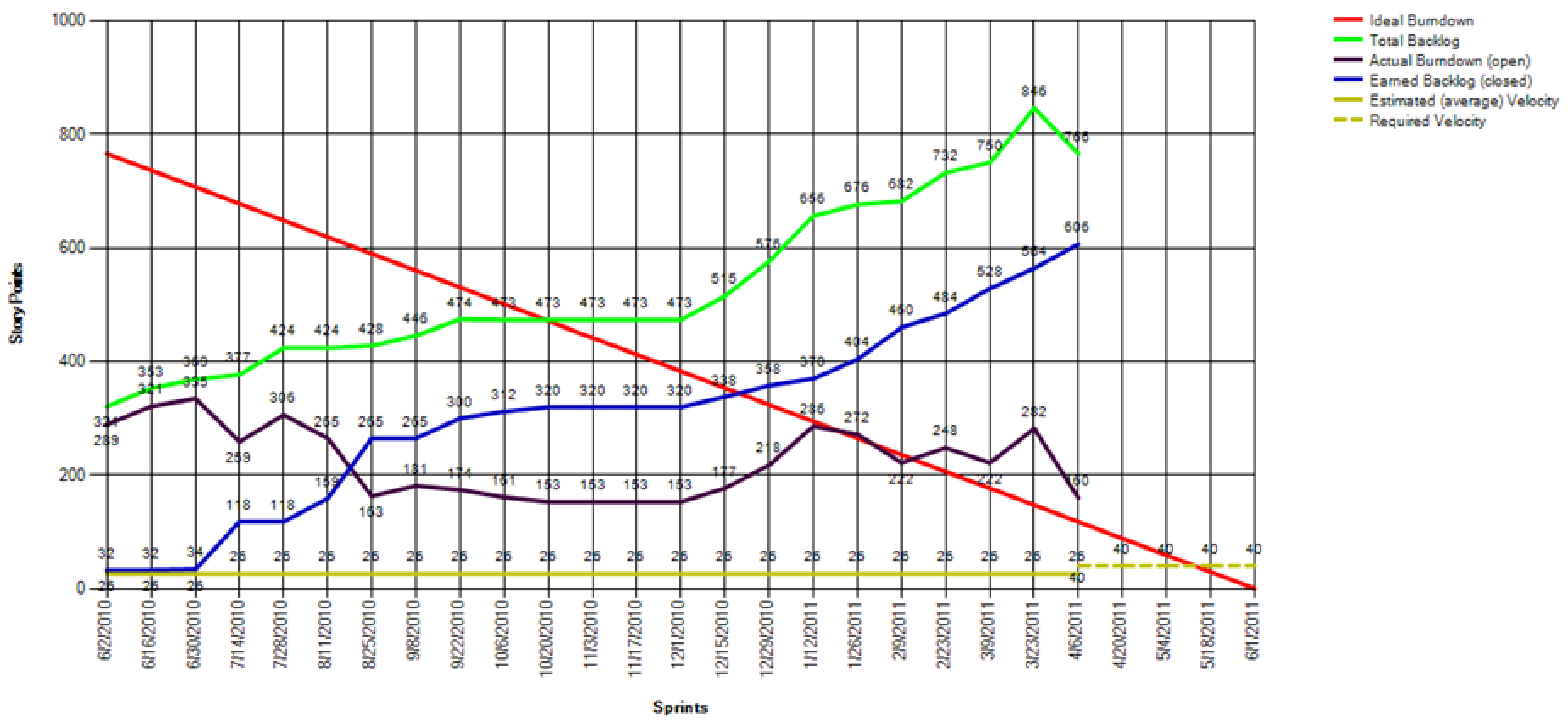
- Burndown charts show the rate at which features are being completed (burned down)
- Burndown charts are completed at iteration as well as release level (and look the same)
- Point in time measurement of amount of work left to be done

  Will fluctuate as work is added / removed



© 2011 VersionOne

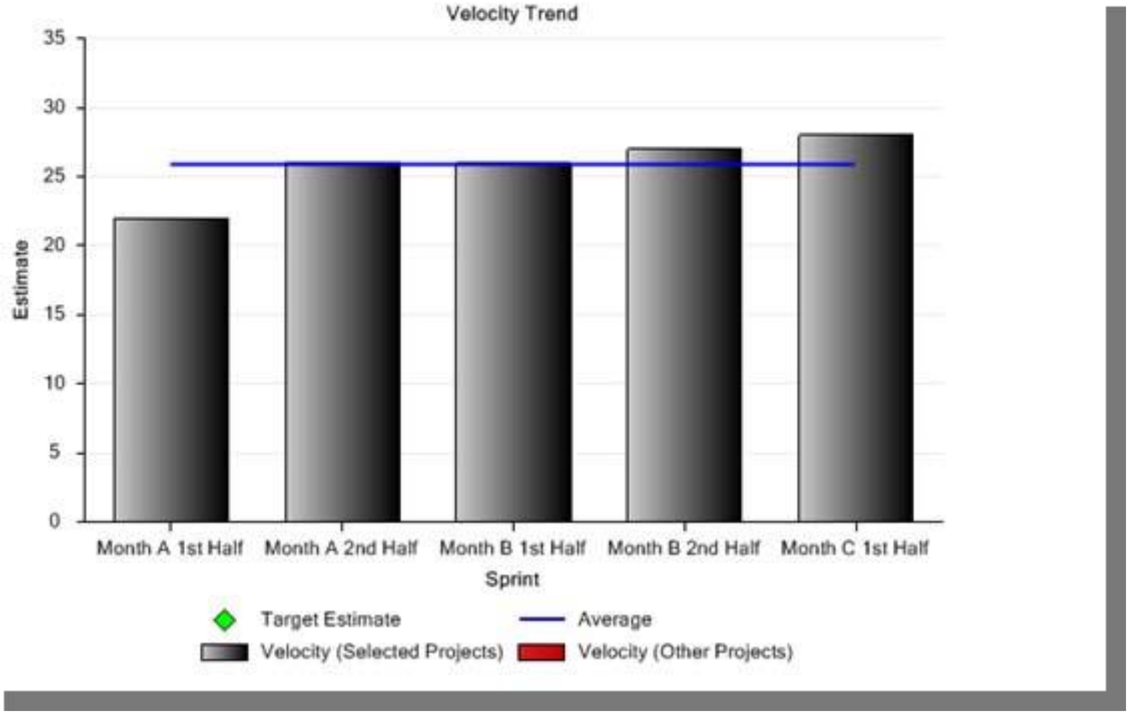
Burndown



iPhone Release 1.0 - Burndown Chart

4/14/2011

Velocity



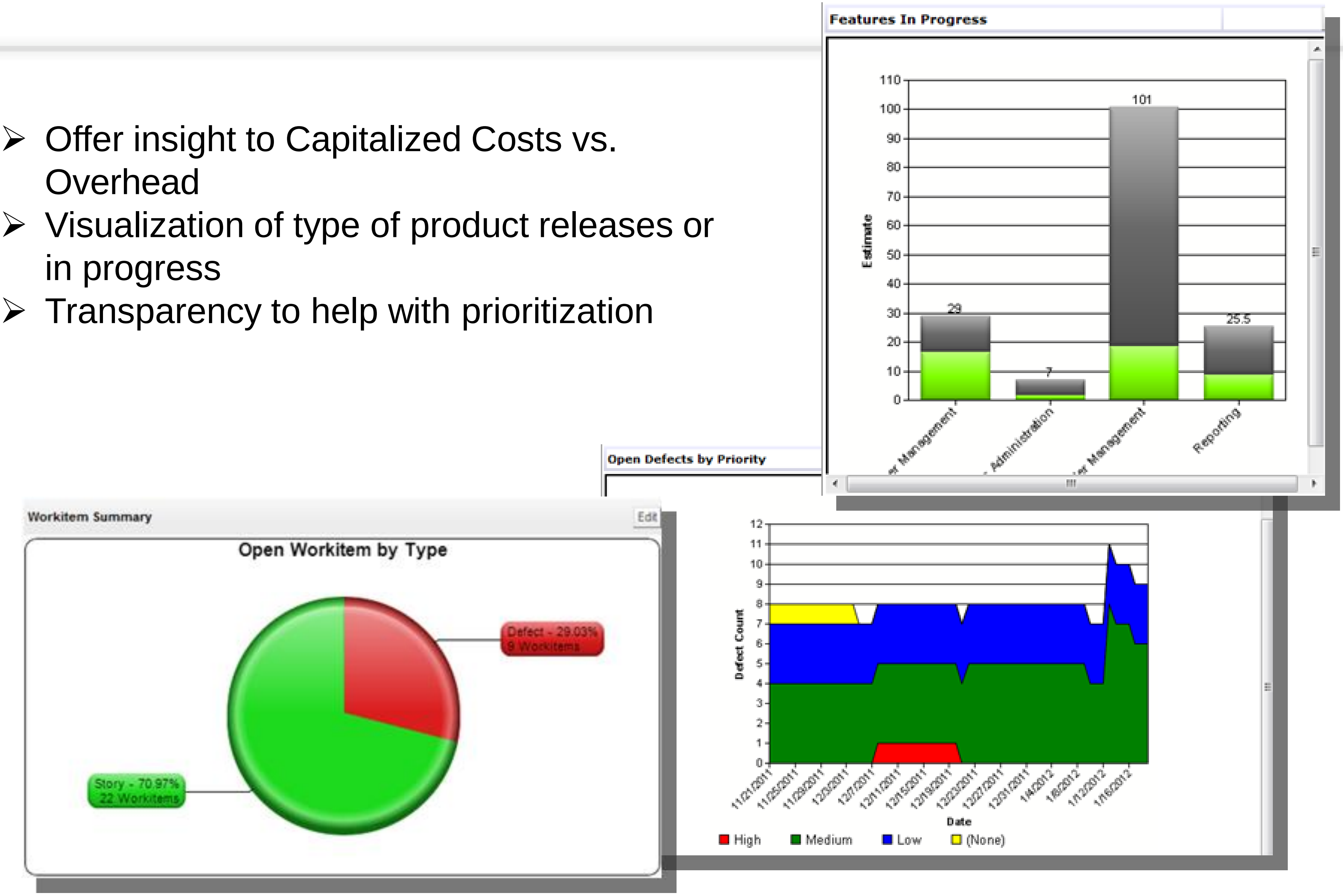
- The **rate** at which a team can produce working software

- More accurately stated, it is measured in terms of the ***stabilized number of [estimation units] a team can deliver per sprint of a given length, and with a given definition of Done.***

© 2011 VersionOne

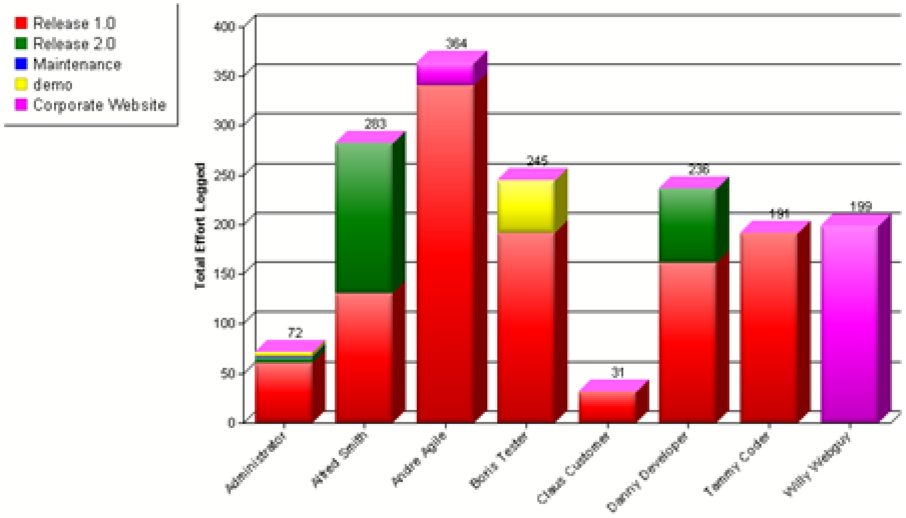
Work Item Counts

- Offer insight to Capitalized Costs vs. Overhead
- Visualization of type of product releases or in progress
- Transparency to help with prioritization



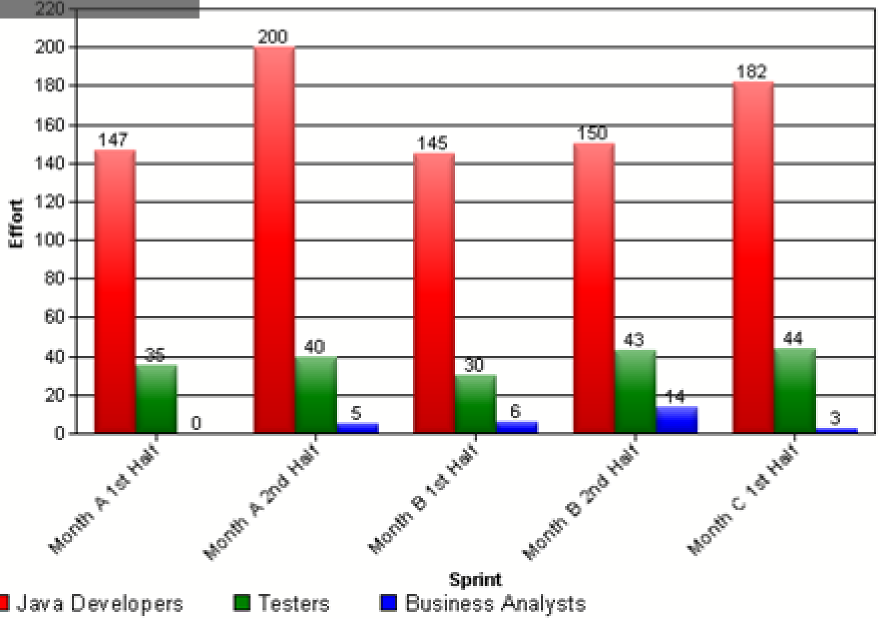
© 2011 VersionOne

Team Member Load/Effort



- ➢ Transparency into a members' load across all projects
- ➢ Helps limit risk
- ➢ Visibility for teams to plan ad hoc depending on capacity adjustments

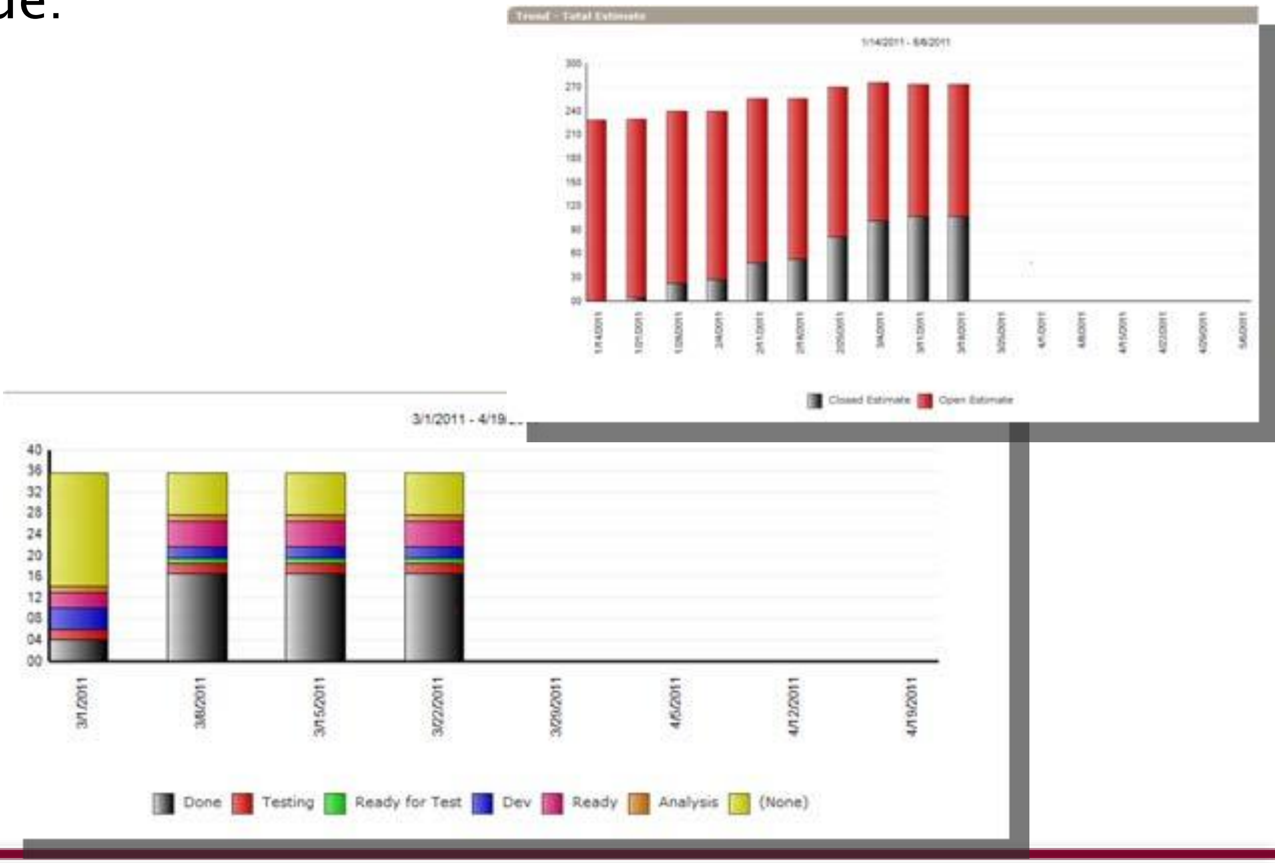
- ➢ Projections for people allocation
- ➢ Helps with budgeting



© 2011 VersionOne

Trend Analysis

- ➢ Trend reports display general trends and changes over the course of time. Use trend reports to understand the differences introduced over the course of time
- ➢ Trend Reports Include:
  - ➢ Estimate Trend
  - ➢ Cumulative Flow
  - ➢ Detail Estimate
  - ➢ Test Status
  - ➢ Issues
  - ➢ Request Status
  - ➢ Defect Status
  - ➢ Velocity Trend
  - ➢ Member Load



© 2011 VersionOne

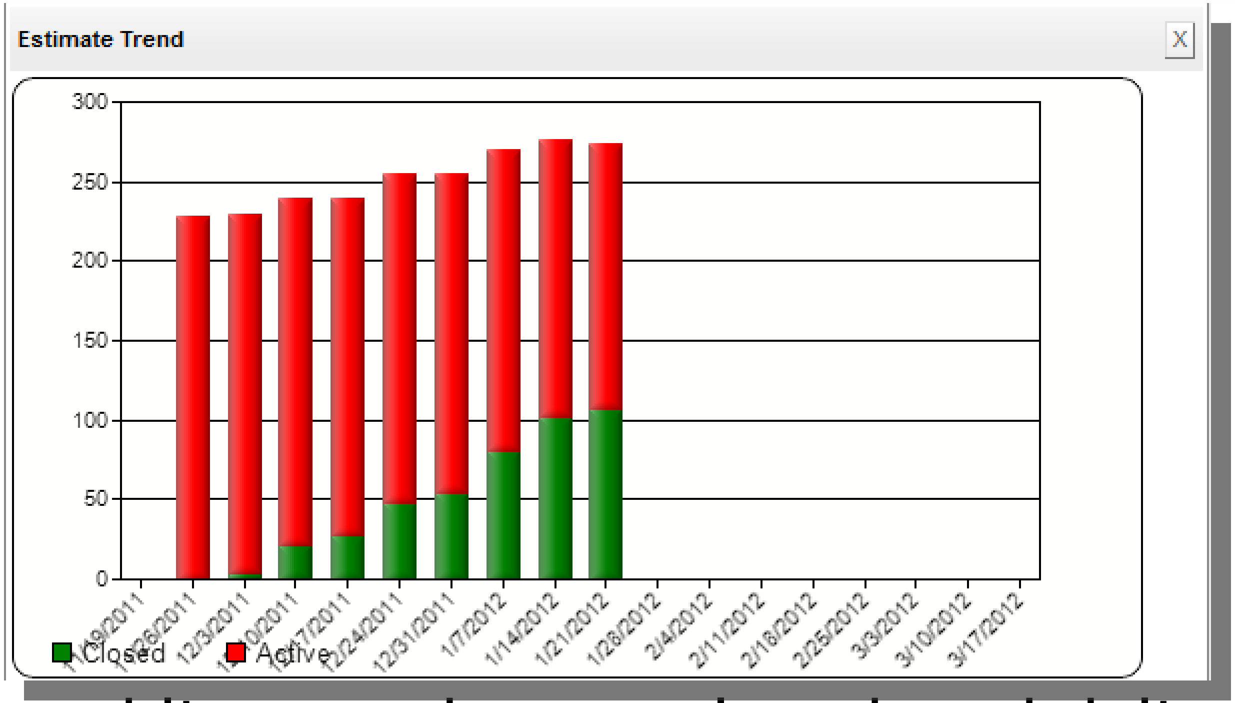
Burnup

- Burnup charts are similar to burndown charts but with total work and completion as separate data points
- Burnup charts are completed at iteration as well as release level (and look the same)
- Point in time measurement of amount of work left to be done

  Will fluctuate as work is added / removed

© 2011 VersionOne

Burnup (Estimate Trend)

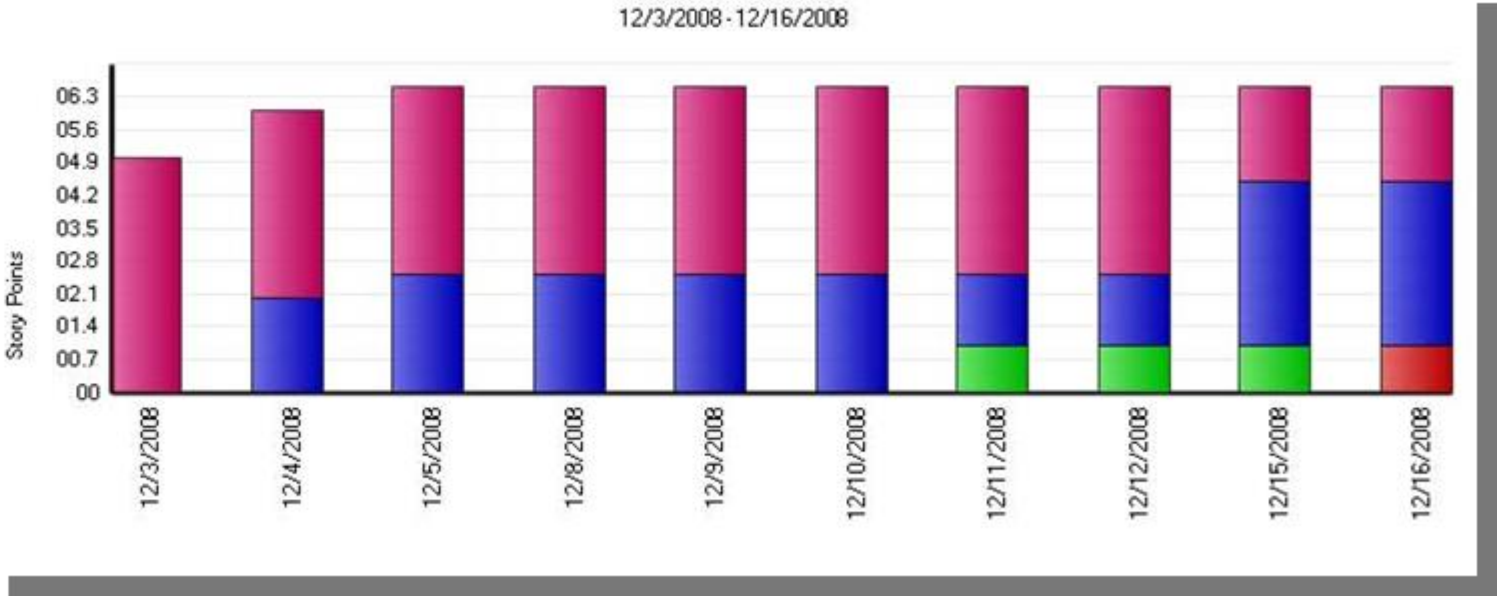


- Use trend lines to show total work and delivery projections
- Used by product owners to help with scope / date decisions as well as delivery team to see if they need cut items in a sprint

© 2011 VersionOne

Cumulative Flow

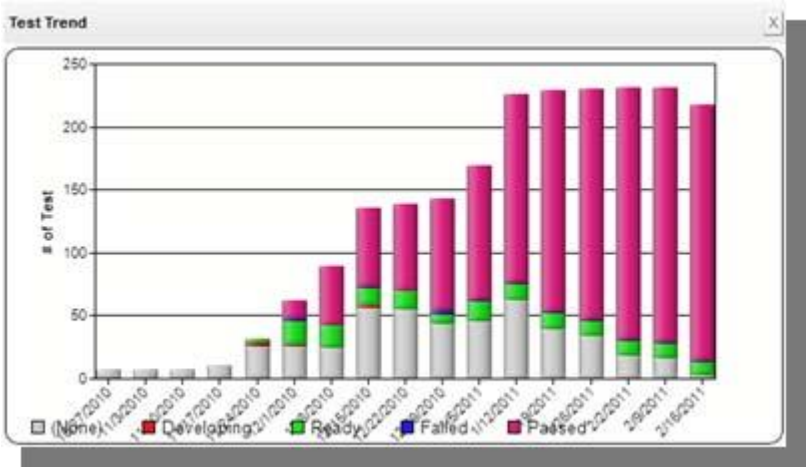
- The Cumulative Flow trend breaks out story and defect estimate by status and tracks that over the course of time within the selected project.
- Use this graph to track the amount of estimate that is in each status as teams work.

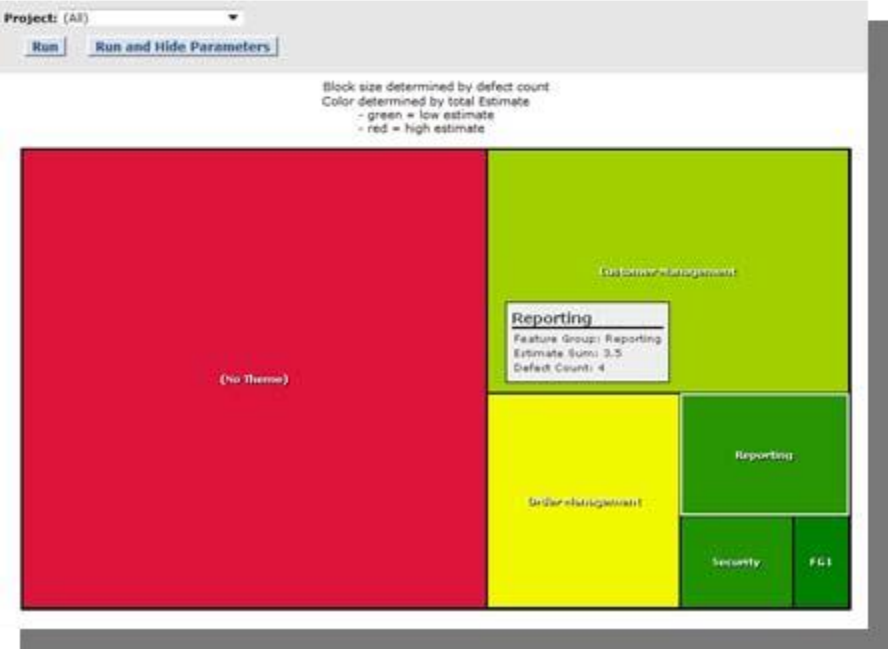


© 2011 VersionOne

Tests and Defects

- ➢ How good is the quality we are developing?
- ➢ Do we need help in testing?
- ➢ Are we automating enough? Is it EVER Enough?
- ➢ Do we need to ask for testing help?




Project: (All)

Block size determined by defect count
Color determined by total Estimate
- green = low estimate
- red = high estimate

Reporting
Feature Group: Reporting
Estimate Sum: 3.5
Defect Count: 4

© 2011 VersionOne