Managing Flow in Agile Projects

Chuck Cobb

AGILE PROJECT MGT AUTHOR AND INSTRUCTOR

@chuckcobb3 www.managedagile.com

Module Overview

Kanban Overview/Examples

Kanban Boards

Flow and the Theory of Constraints

Cumulative Flow Diagrams

Kanban Overview

What Is Kanban?

“Kanban is a visual signal that’s used to trigger an action.

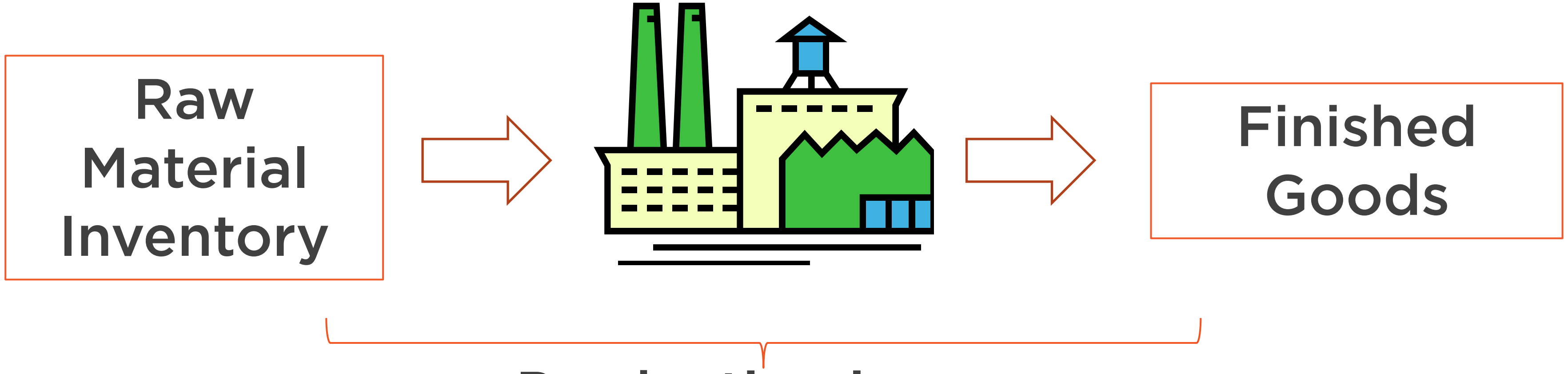
The word Kanban is Japanese. Roughly translated, it means ‘card you can see’.”

http://whatis.techtarget.com/definition/kanban

Differences Between "Push" and "Pull" Systems

A "push" system is totally planned in advance and production capacity is scheduled against a long-term forecast

Example:



Raw Material Inventory

Finished Goods

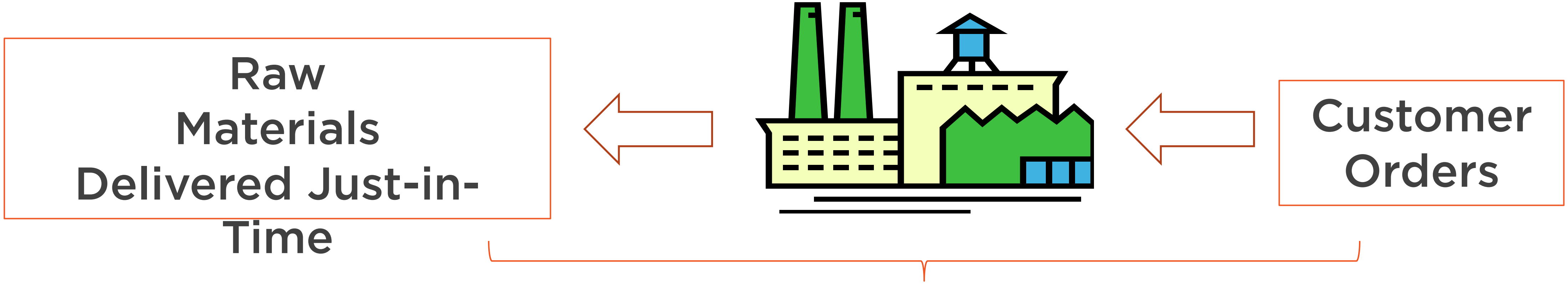
Production is Planned Against a Forecast

Waterfall is a “push” system – An attempt is made to plan all of the requirements in advance and “push” them through the development process

Differences Between “Push” and “Pull” Systems

A “pull” system is totally driven by demand and built to satisfy immediate actual needs (not forecasted needs based on a plan)

Example:



Raw Materials Delivered Just-in-Time

Customer Orders

Production is Scheduled to Fill Customer Demand

Kanban is a “Pull” System – It is Totally Driven by Demand Rather Than Being Planned

Differences Between Scrum and Kanban

Scrum is a push/pull system

Work is planned to some extent in the product backlog and broken up into sprints and releases

Within a sprint, the work is planned, limited, and allocated to the team

However, it is demand-driven based on priorities set by the product owner

Kanban is totally a pull system

No attempt is made to plan requirements in advance

A Kanban process does not break up the work into sprints at all

It is a continuous flow model – work is taken into the process immediately based on priority as soon as resources are available to work on it

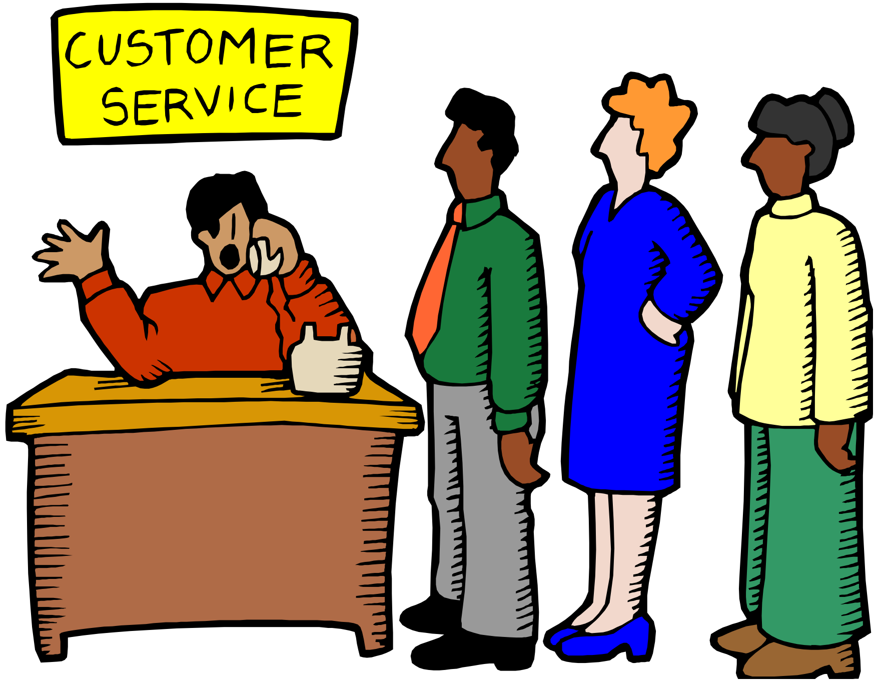
Cobb, Charles, Managed Agile Development, Outskirts Press, 2013

Kanban Example – Customer Service Queue

Some number of customer service agents are trained to respond to customer support issues

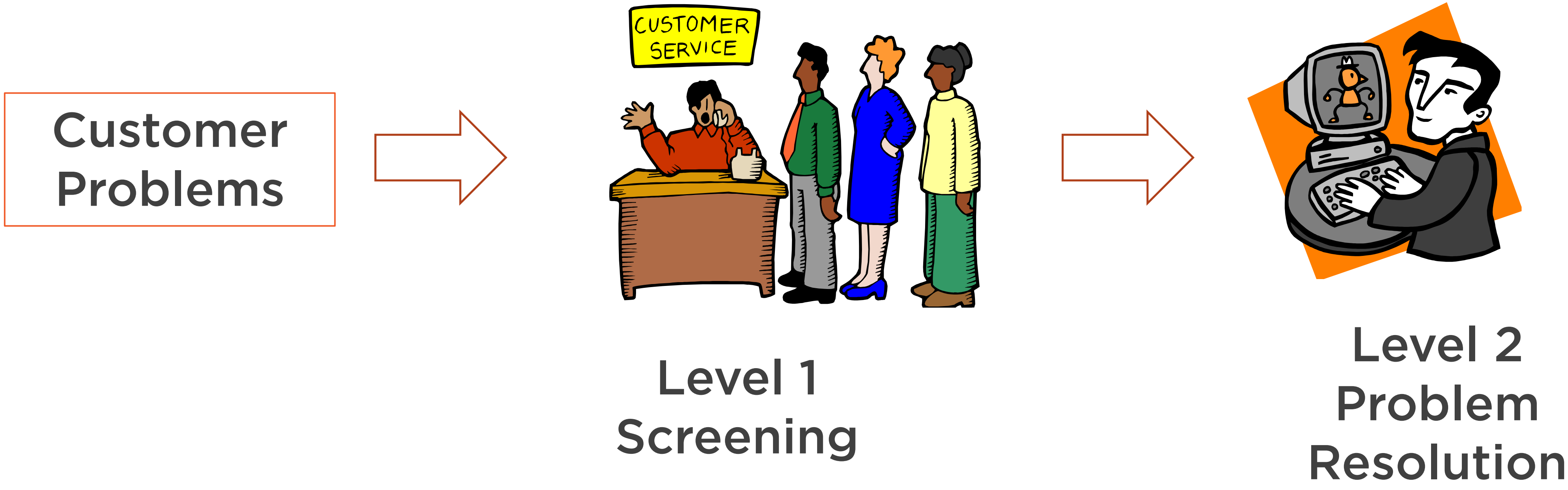
It is totally demand-driven by customers calling in with issues

It is impossible to completely plan all aspects of the process



Work in Process (WIP) Limits in Kanban

Kanban Processes Many Times Have Stages



Customer Problems

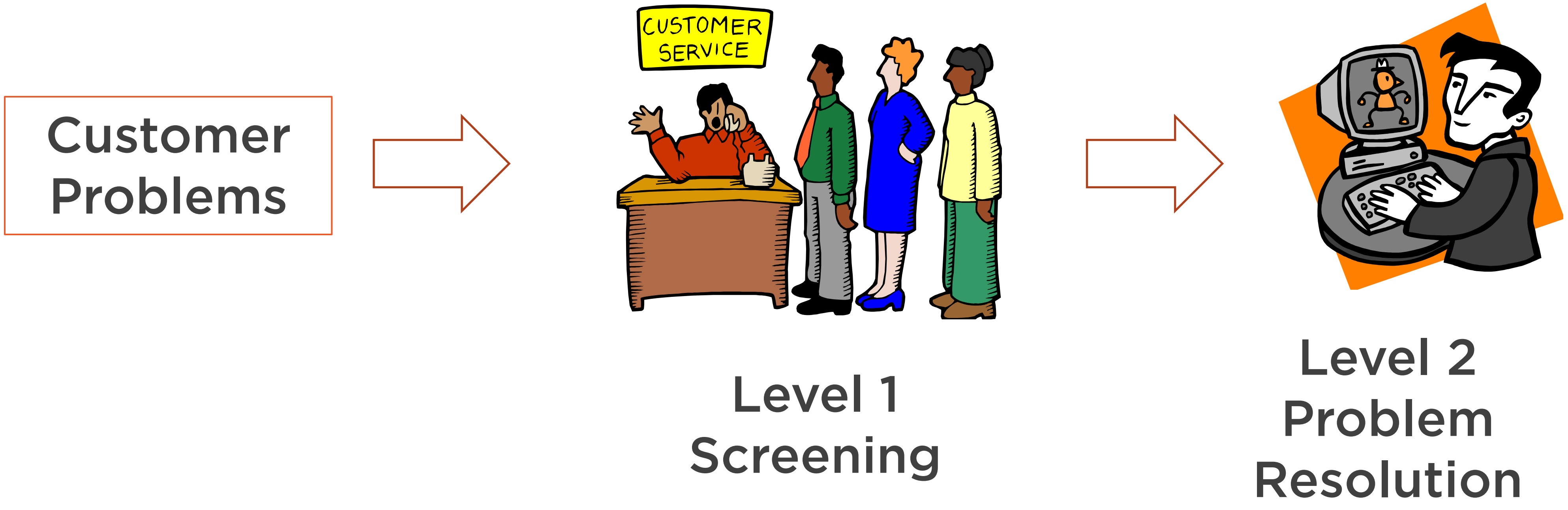
Level 1 Screening

Level 2 Problem Resolution

Work in Process (WIP) Limits in Kanban

The process is designed to optimize overall flow

Any individual stage in the process may have capacity limitations to limit overall “work in process” or flow



Customer Problems

Level 1 Screening

Level 2 Problem Resolution

Kanban Examples

Kanban Process Examples - Reports

A demand-driven process to produce reports

A development group is assigned to produce business reports for users

Work is unplanned and demand-driven

Kanban Process Examples – Mature Products

A product has reached a mature stage in its lifecycle that calls for ongoing support

Feature requests and bug support tasks are queued on the basis of priority

Work is performed on a continuous basis as resources are available to work on it

Kanban Process Examples – Unplanned Projects

Small and simple projects that do not require significant planning

A Kanban process might be more streamlined and more efficient,

But of course, the caveat is that it is probably totally unplanned

Kanban is a Foundation for Scrum

Within a sprint in Scrum, a Kanban process is used to manage the flow of work-in-process

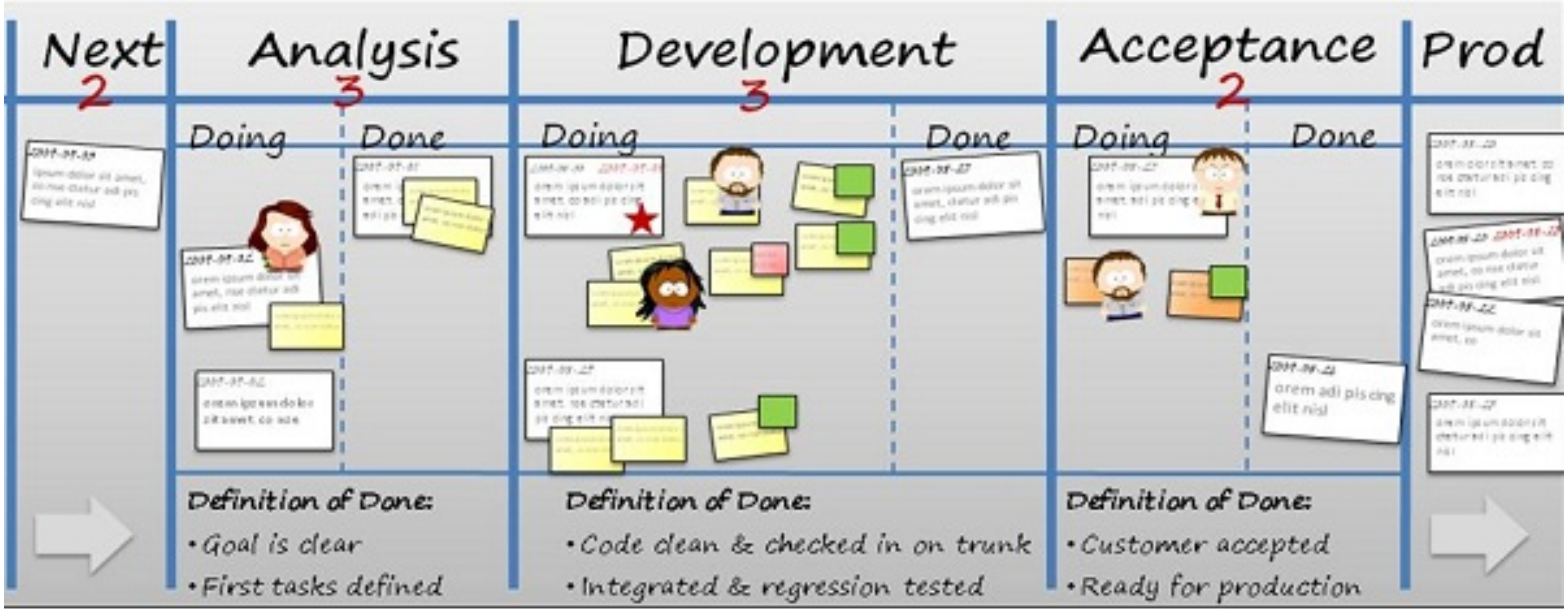
TO DO
IN PROGRESS
COMPLETED

Kanban Flow

Kanban Boards

Kanban Boards

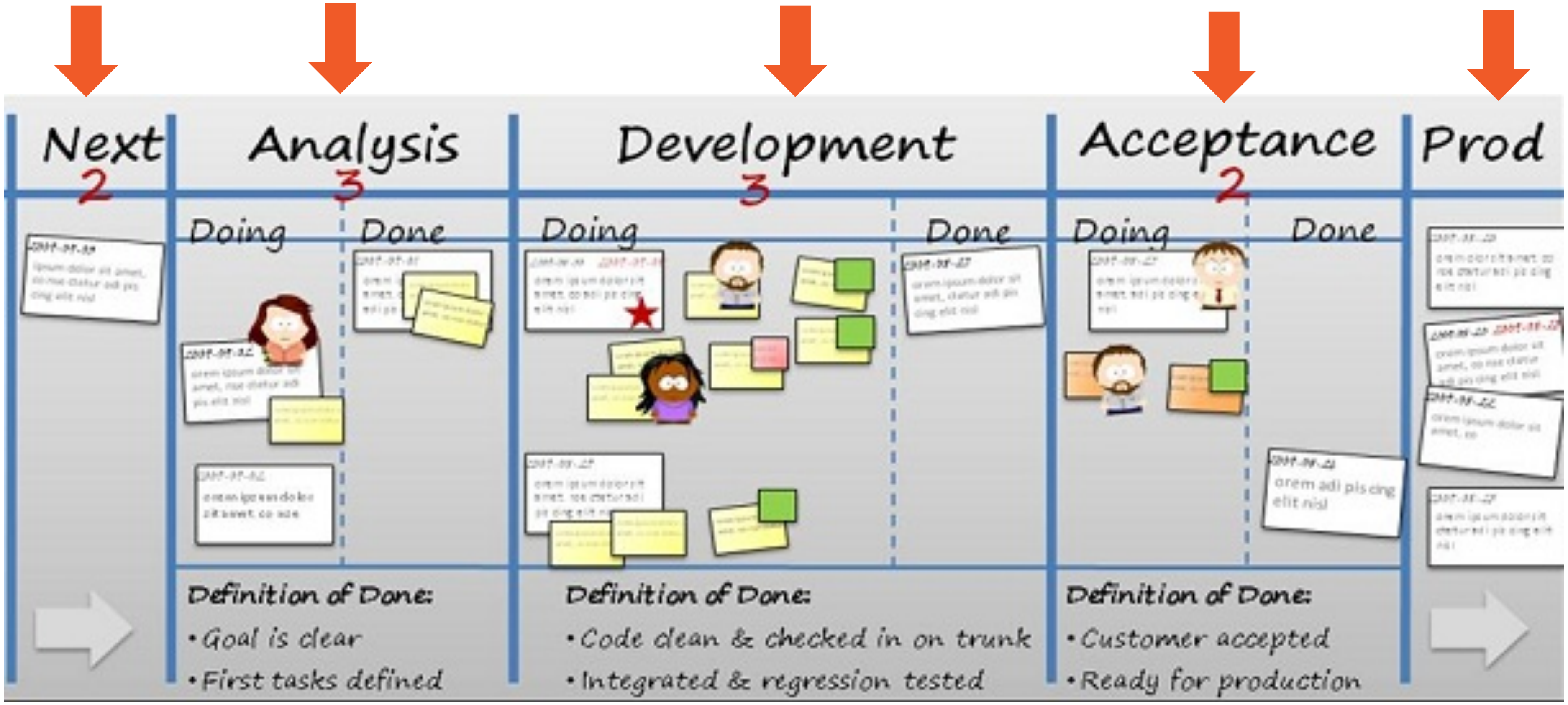
Kanban Boards Are Used to Visually Show the Flow of Items Through a Kanban Process



Kanban and Scrum – Making the Most of Both – http://www.open-ware.org/eng/news/kniberg2.htm

Kanban Boards

Columns in a Kanban Board Represent Stages in the Kanban Process



Definition of Done:
- Goal is clear
- First tasks defined

Definition of Done:
- Code clean & checked in on trunk
- Integrated & regression tested

Definition of Done:
- Customer accepted
- Ready for production

Kanban and Scrum – Making the Most of Both –
http://www.open-ware.org/eng/news/kniberg2.htm

Kanban Boards with “Stickies”

Simple Kanban boards can be done with “White Boards” and “stickies”

TO DO
IN PROGRESS
COMPLETED

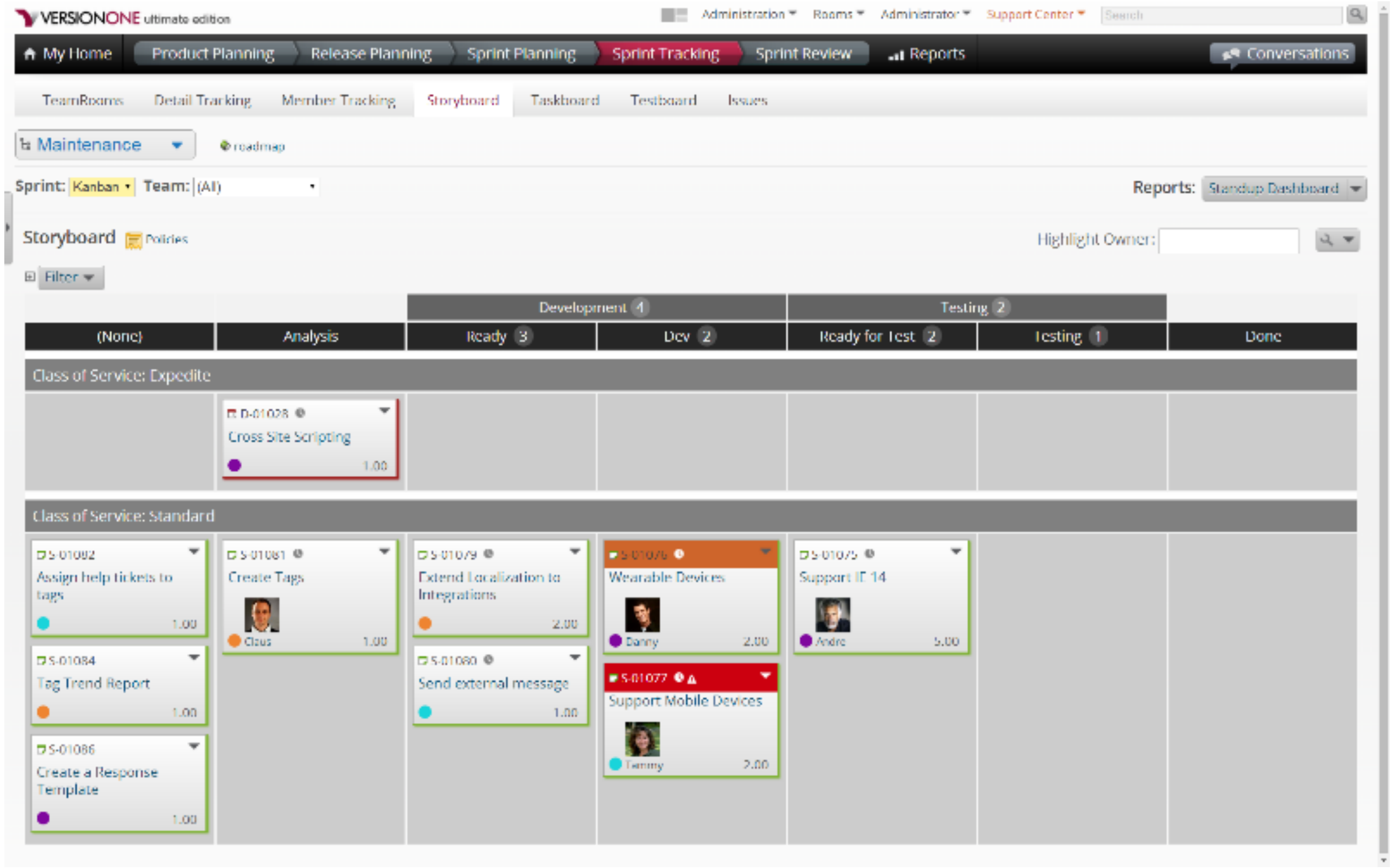
Kanban Boards with “Stickies”

Limitations:

- The data on the board is not updated automatically as progress is made.
- Doesn’t work well for distributed teams
- Also doesn’t provide a management reporting capability

Kanban Boards with Online Tools

Kanban Boards Can Be Done With an Online Tool



Sprint: Kanban Team: (All)

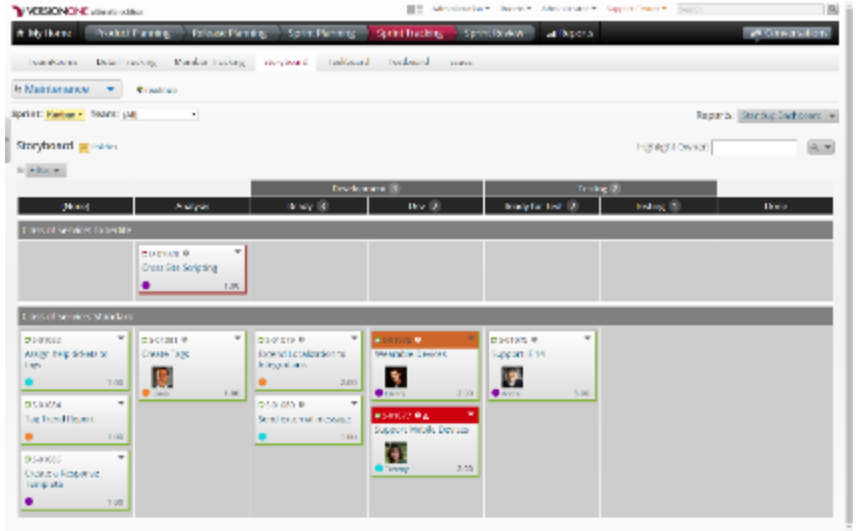
Reports: Standup Dashboard

		Development 4		Testing 2		
(None)	Analysis	Ready 3	Dev 2	Ready for Test 2	Testing 1	Done
Class of Service: Expedite						
	D-01028 Cross Site Scripting 1.00					
Class of Service: Standard						
S-01082 Assign help tickets to tags 1.00 S-01084 Tag Trend Report 1.00 S-01086 Create a Response Template 1.00	S-01081 Create Tags Claus 1.00	S-01079 Extend Localization to Integrations 2.00 S-01080 Send external message 1.00	S-01076 Wearable Devices Danny 2.00 S-01077 Support Mobile Devices Tammy 2.00	S-01075 Support IE 14 Andre 5.00		

Kanban Boards with Online Tools

Advantages:

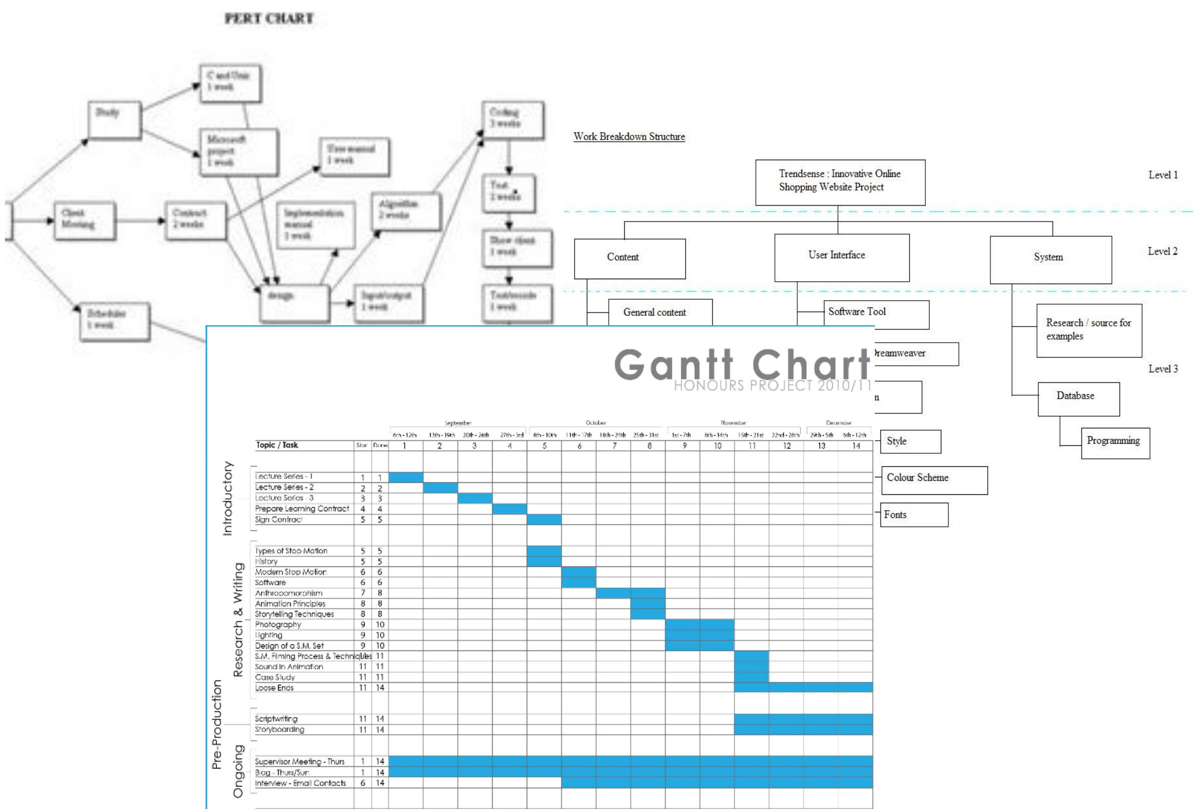
- The data can be automatically updated in real-time as the work is being done
- Anyone can have direct access to either view or update the information locally or remotely
- The information on the board can automatically feed reporting tools such as burndown charts



Principles of Product Development Flow

Why is the Concept of “Flow” Important?

Traditional Project



Work Breakdown Structure

Trendsense : Innovative Online Shopping Website Project

Level 1

Content

User Interface

System

Level 2

General content

Software Tool

reamweaver

n

Style

Colour Scheme

Fonts

Research / source for examples

Database

Programming

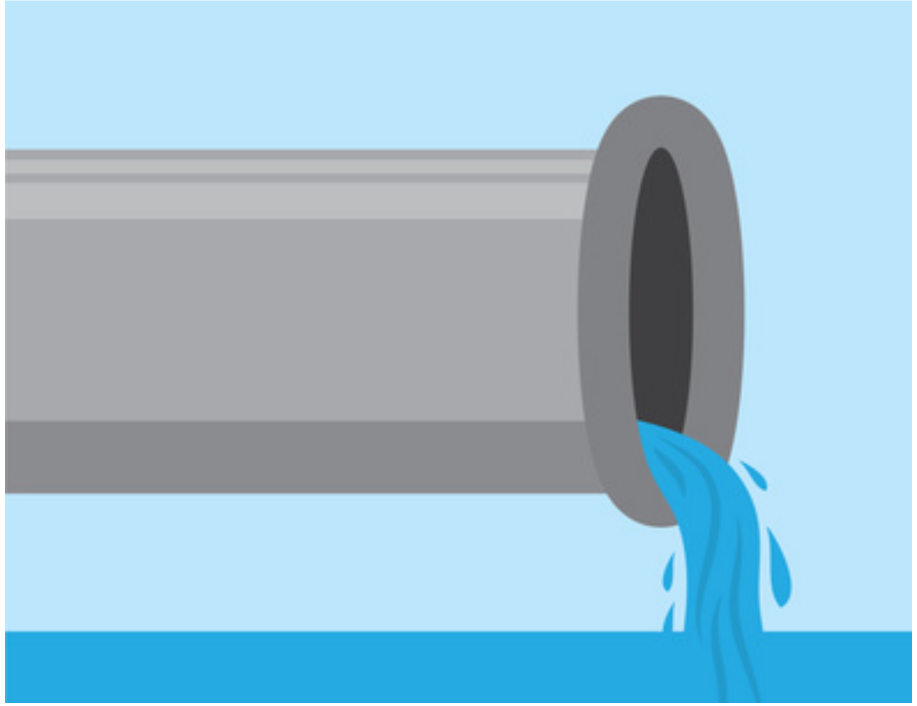
Level 3

Gantt Chart

HONOURS PROJECT 2010/11

				September				October				November				December	
				6th - 12th	13th - 19th	20th - 26th	27th - 3rd	4th - 10th	11th - 17th	18th - 24th	25th - 31st	1st - 7th	8th - 14th	15th - 21st	22nd - 28th	29th - 5th	6th - 12th
	Topic / Task	Star	Done	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Introductory	Lecture Series - 1	1	1	🟦													
	Lecture Series - 2	2	2		🟦												
	Lecture Series - 3	3	3			🟦											
	Prepare Learning Contract	4	4				🟦										
	Sign Contract	5	5					🟦									
Research & Writing	Types of Stop Motion	5	5					🟦									
	History	5	5					🟦									
	Modern Stop Motion	6	6						🟦								
	Software	6	6						🟦								
	Anthropomorphism	7	8							🟦	🟦						
	Animation Principles	8	8								🟦						
	Storytelling Techniques	8	8								🟦						
	Photography	9	10									🟦	🟦				
	Lighting	9	10									🟦	🟦				
	Design of a S.M. Set	9	10									🟦	🟦				
	S.M. Filming Process & Techniques	11	11											🟦			
	Sound In Animation	11	11											🟦			
	Case Study	11	11											🟦			
	Loose Ends	11	14											🟦	🟦	🟦	🟦
Pre-Production	Scriptwriting	11	14											🟦	🟦	🟦	🟦
	Storyboarding	11	14											🟦	🟦	🟦	🟦
Ongoing	Supervisor Meeting - Thurs	1	14	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
	Blog - Thurs/Sun	1	14	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
	Interview - Email Contacts	6	14						🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦

Agile Project



Principles of Product Development Flow

1. Take an Economic View

Prioritize requirements by highest business value first

Defer lower value requirements

Recognize the point of diminishing returns and develop the Minimum Viable Product first

Reinertsen, Don, The Principles of Product Development Flow, Celeritas Publishing, 2009

Principles of Product Development Flow

2. Queues – Actively Manage Queues

Having too many items waiting in a queue to start development is wasteful and inefficient

The requirements may change prior to going into development

Speculation in the requirements that are done too far into the future can result in erroneous assumptions

Reinertsen, Don, The Principles of Product Development Flow, Celeritas Publishing, 2009

Principles of Product Development Flow

3. Variability – Understand and Exploit Variability

Breaking up large requirements into smaller ones that are of a more uniform size reduces variability and can improve flow

4. Batch Size – Reduce Batch Size

Large batch sizes tend to cause bottlenecks and inhibit flow

Reinertsen, Don, The Principles of Product Development Flow, Celeritas Publishing, 2009

Principles of Product Development Flow

5. WIP Constraints – Apply Work in Progress (WIP) Constraints

Don’t take on more work than you can handle

Be aware of capacity limitations in different stages of work

Use specialized resources wisely

Reinertsen, Don, The Principles of Product Development Flow, Celeritas Publishing, 2009

Principles of Product Development Flow

6. Control Flow Under Uncertainty – Cadence and Synchronization

Having a repeatable cadence improves the efficiency of the process and

Allows synchronizing a predictable development process with a much more unpredictable flow of requirements

Reinertsen, Don, The Principles of Product Development Flow, Celeritas Publishing, 2009

Principles of Product Development Flow

7. Fast Feedback – Get Feedback as Fast as Possible

Fast feedback can lower the expected loss by truncating unproductive paths more quickly or

Raise the expected gain because we can exploit an emergent opportunity by rapidly redirecting resources

Reinertsen, Don, The Principles of Product Development Flow, Celeritas Publishing, 2009

Principles of Product Development Flow

8. Decentralized Control – Decentralize Control

Decentralizing control enables much more rapid and more effective decision-making

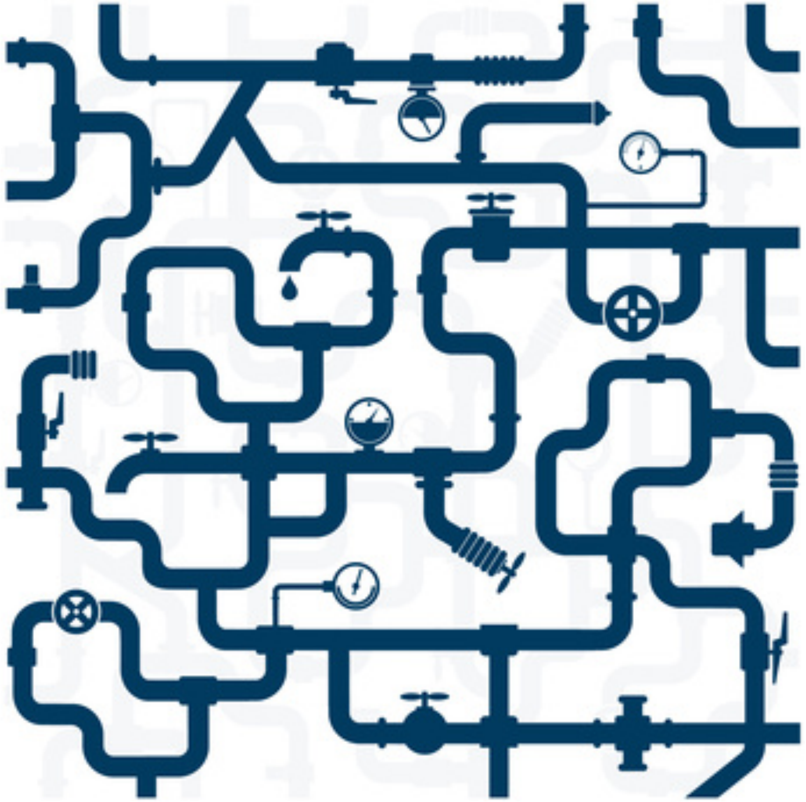
Reinertsen, Don, The Principles of Product Development Flow, Celeritas Publishing, 2009

Theory of Constraints

Theory of Constraints

The “Theory of Constraints” is a systematic method for analyzing and improving process flow

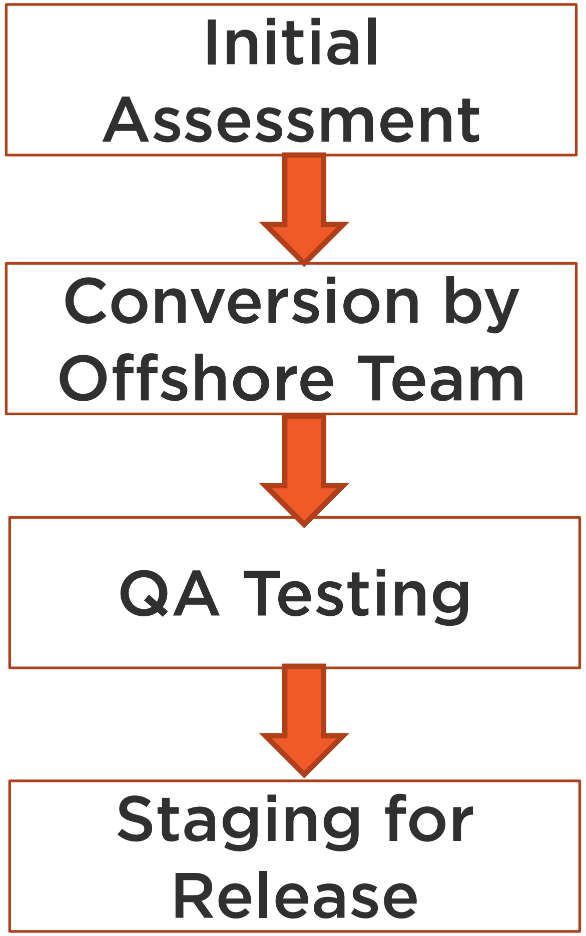
The Theory of Constraints was originally developed by Elihu Goldratt and was originally published in his book, “The Goal”



Goldratt, Eliyu, The Goal: A Process of Ongoing Improvement, North River Press, June 2012

Theory of Constraints - Example

Process designed to upgrade over 100 applications to a new version of Visual Studio.Net:



Initial Assessment

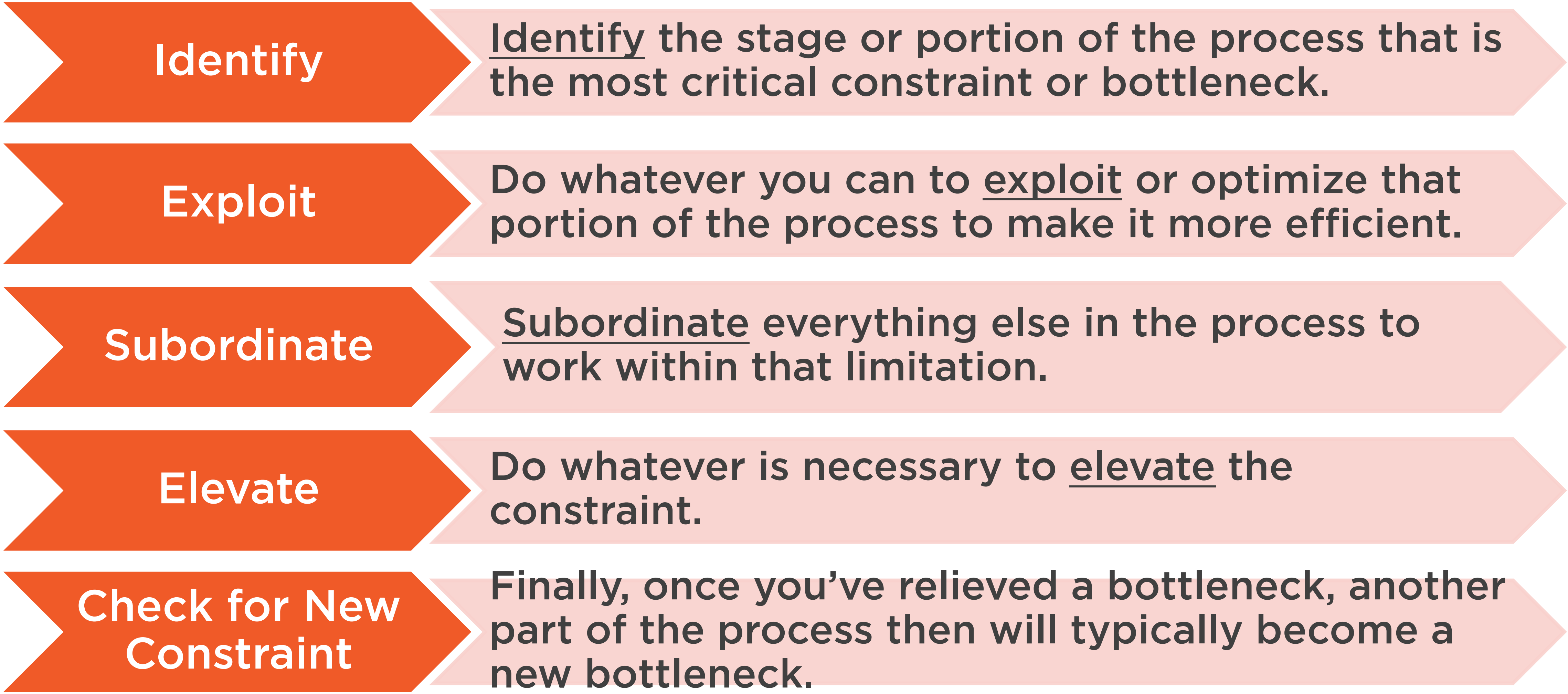
Conversion by Offshore Team

QA Testing

Staging for Release

Cobb, Charles, Managed Agile Development – Making Agile Work for Your Business, 2013 p 42-43

Theory of Constraints



Identify: Identify the stage or portion of the process that is the most critical constraint or bottleneck.

Exploit: Do whatever you can to exploit or optimize that portion of the process to make it more efficient.

Subordinate: Subordinate everything else in the process to work within that limitation.

Elevate: Do whatever is necessary to elevate the constraint.

Check for New Constraint: Finally, once you've relieved a bottleneck, another part of the process then will typically become a new bottleneck.

Goldratt, Eliyu, The Goal: A Process of Ongoing Improvement, North River Press, June 2012

Theory of Constraints – Example

Step	Result
Identify the Primary Constraint	QA Test Resources Was the Primary Constraint on the Overall Process
Exploit the Constraint	Try to Optimize the Use of QA Resources by Spreading the Load

Cobb, Charles, Managed Agile Development –
Making Agile Work for Your Business, 2013 p 42-43

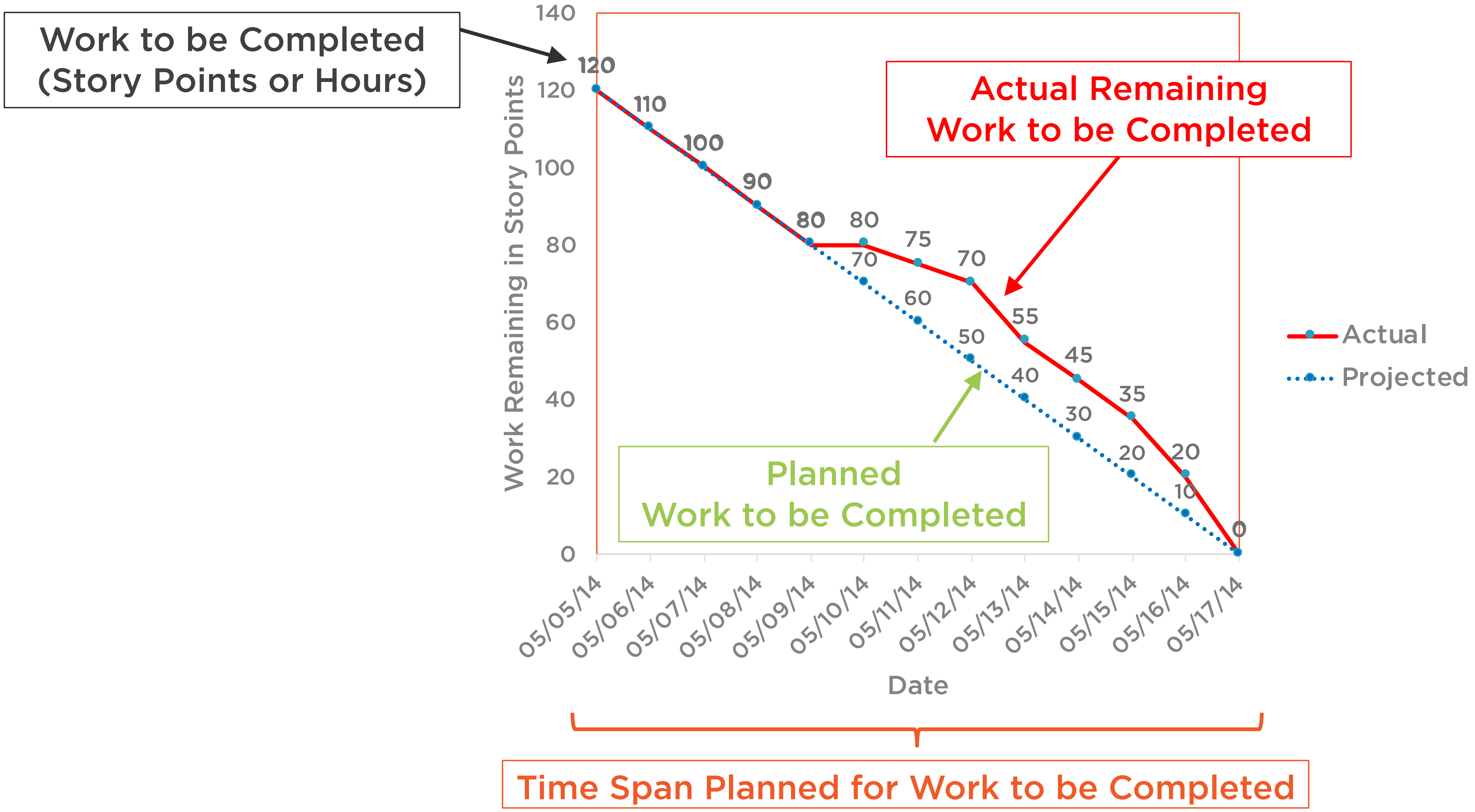
Theory of Constraints – Example

Step	Result
Subordinate Everything Else	Everything Else in the Process Must be Limited by the Primary Constraint (QA Test Resources)
Elevate the Constraint	Add More QA Resources if Possible to Relieve the Constraint
Check for New Constraints	After Adding More QA Resources, Reevaluate the Overall Process and Check for New Constraints

Cobb, Charles, Managed Agile Development –
Making Agile Work for Your Business, 2013 p 42-43

Cumulative Flow Diagrams

What is a Burn-down Chart?



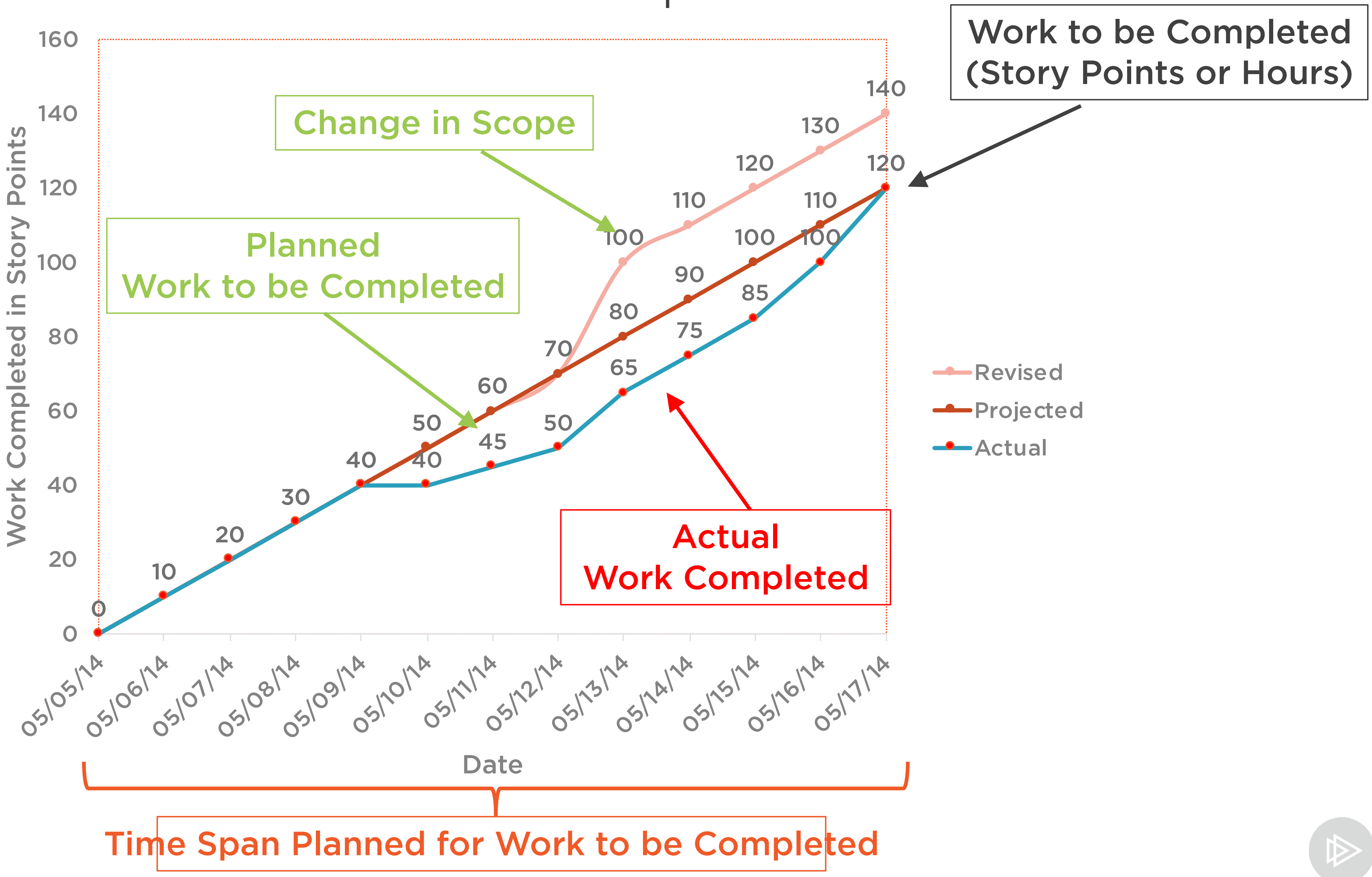
Work to be Completed (Story Points or Hours)

Actual Remaining Work to be Completed

Planned Work to be Completed

Time Span Planned for Work to be Completed

What is a Burn-up Chart?



Change in Scope

Planned Work to be Completed

Work to be Completed (Story Points or Hours)

Actual Work Completed

Time Span Planned for Work to be Completed

Limitations of Burn-down and Burn-up Charts

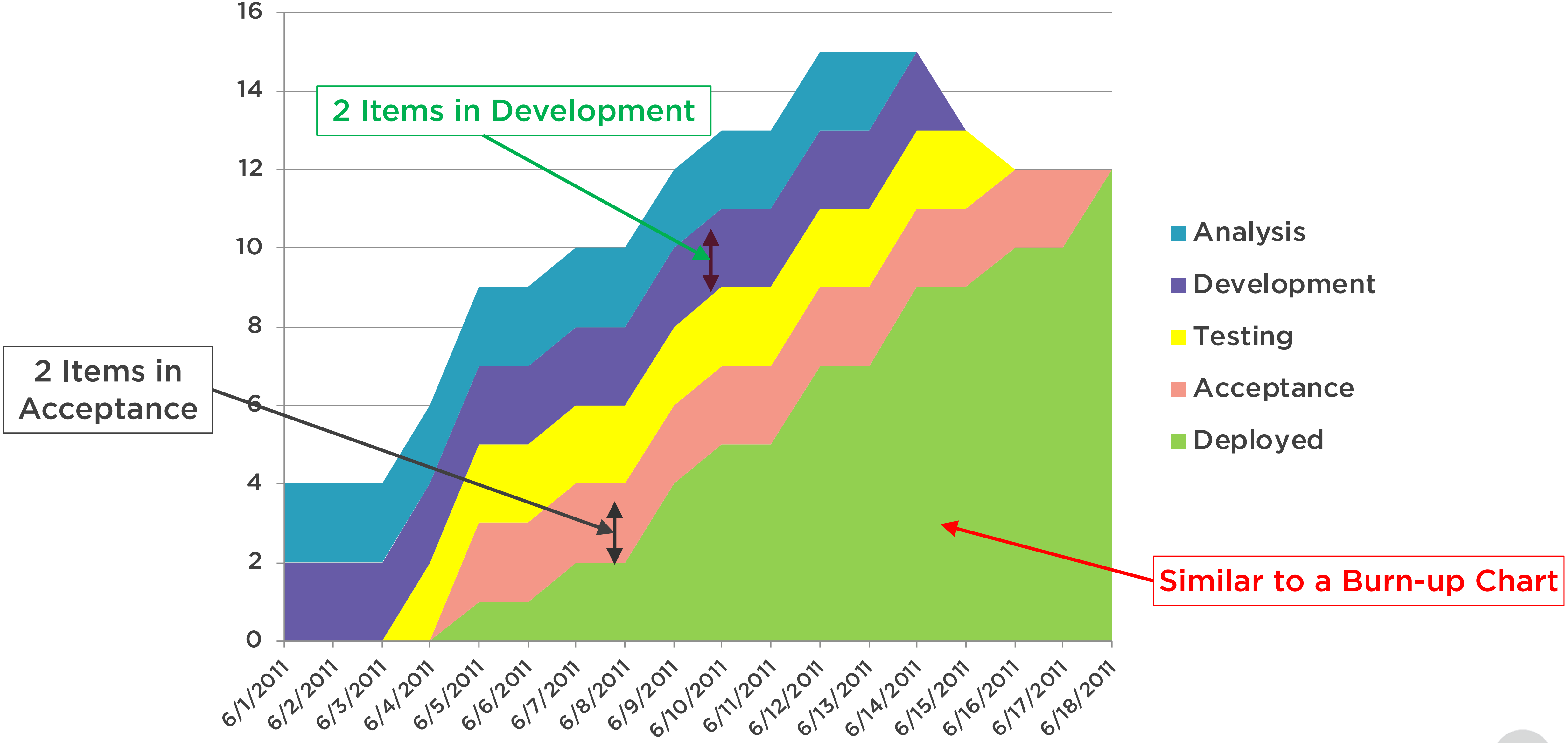
Only shows total work completed or remaining to be completed

Doesn’t provide visibility into:

- How the work is progressing in each stage of the process
- Where the bottlenecks might be

If you want to improve the process flow, you probably need more information

What is a Cumulative Flow Diagram?



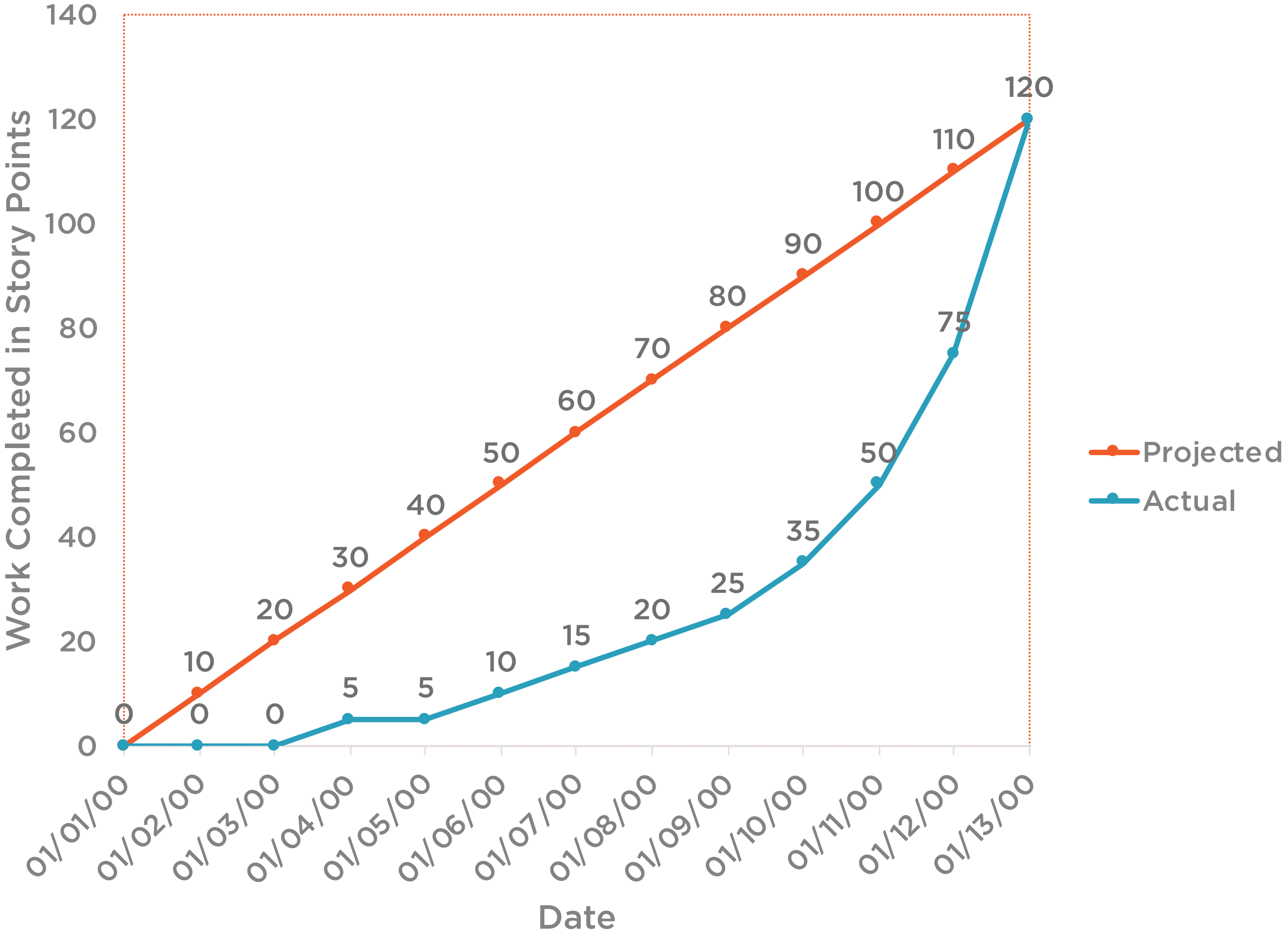
2 Items in Development

2 Items in Acceptance

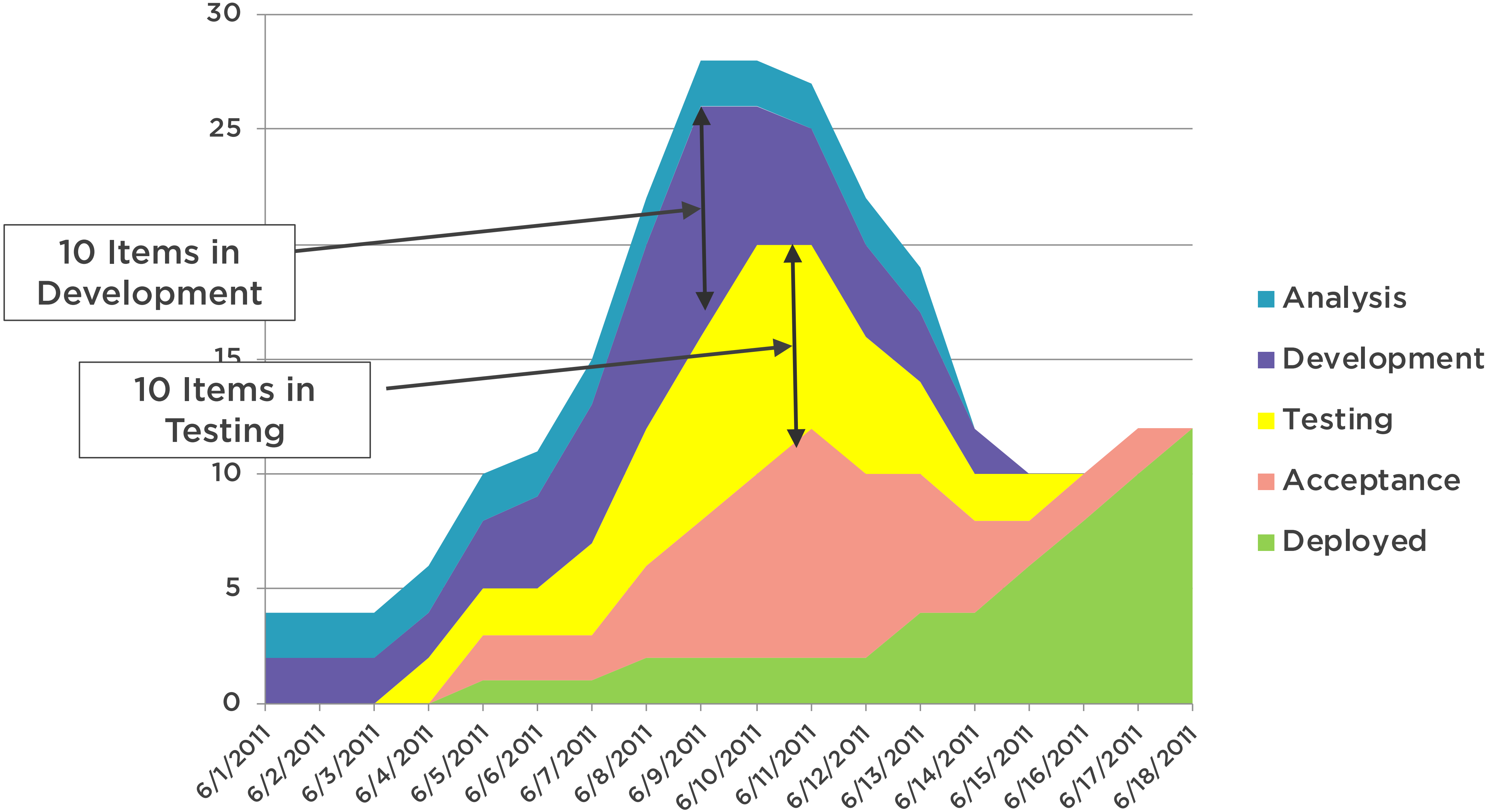
Similar to a Burn-up Chart

Cumulative flow diagrams enable us to measure how efficiently we're delivering valuable, working product to the customer, and indicate where we need to focus our process improvement efforts

What if the Burn-up Was Abnormal?



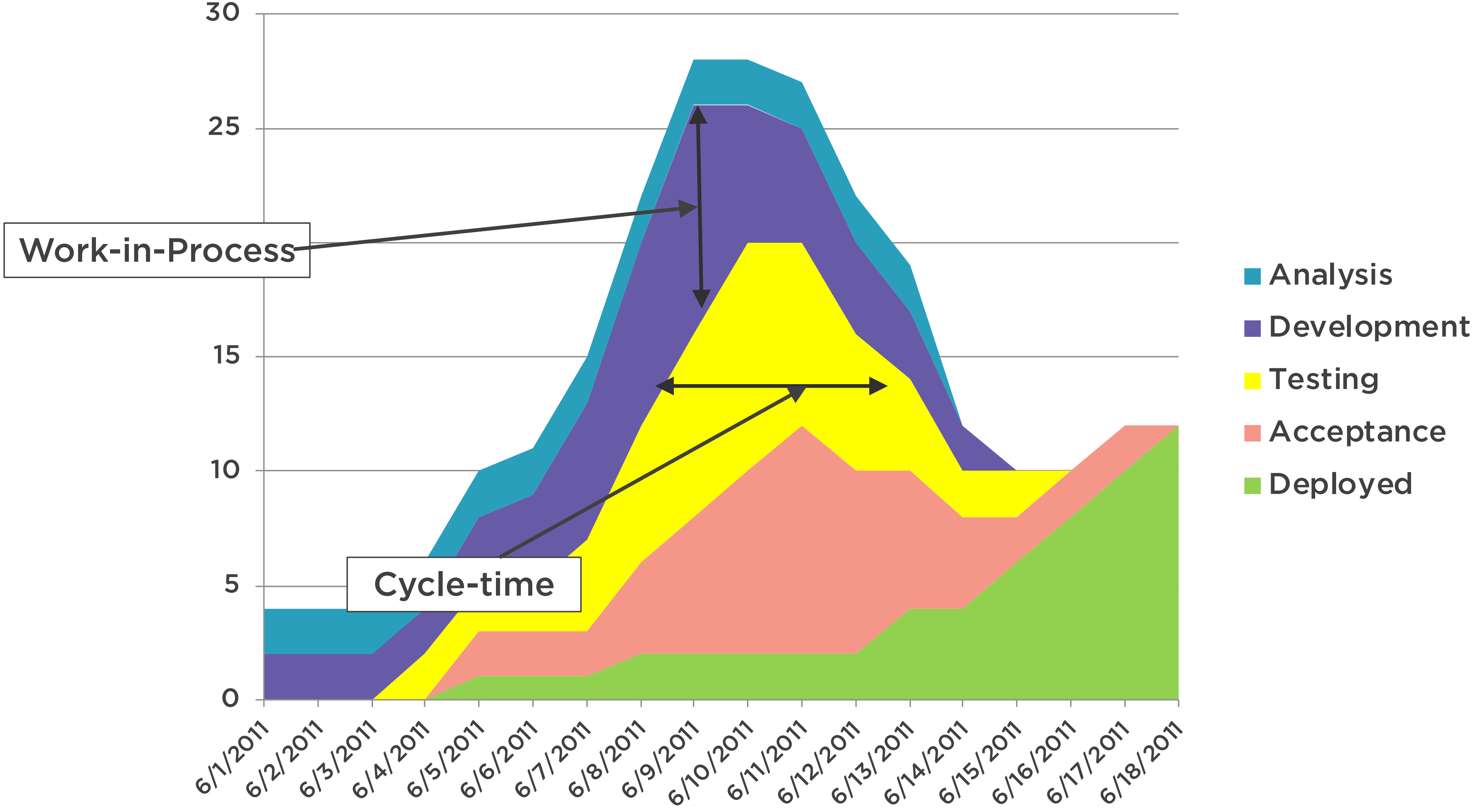
What if the Burn-up Was Abnormal?



10 Items in Development

10 Items in Testing

What Information Does a CFD Provide?



Work-in-Process

Cycle-time

Cumulative flow diagrams are only practical when the team uses an online tool for tracking work

Up Next:
Course Summary