Biography

Robin Yeman
IS&GS Agile Transformation Lead
ASPIRE

Lockheed Martin Business Unit
Email: robin.yeman@lmco.com
Phone: 571-535-5854

Career Highlights : 20 Years at Lockheed Martin, 13 Years of Agile

Roles: *Software Engineer, Systems Engineer, Test Engineer, Capture Management, Engineering Program Manager (EPM), Subcontracts Program Manager (SPM), Program Manager (PM)*

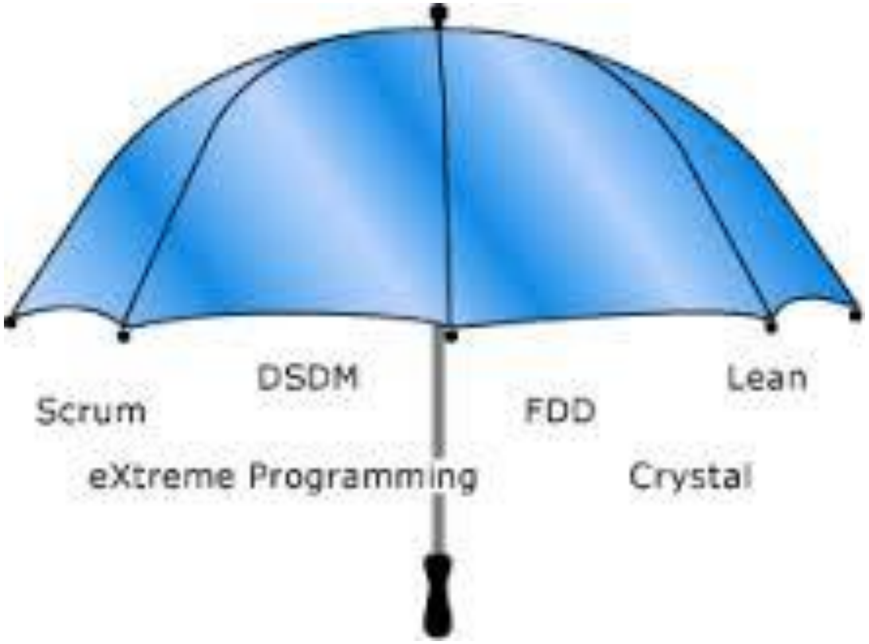
Certifications: *Certified Scrum Master (CSM), Certified Scrum Practitioner (CSP), Professional Scrum Master (PSM), Scaled Agile Program Consultant (SPC), Certified Systems Engineer (CSEP), Program Management Professional (PMP), Program Management Agile Professional (PMI-ACP), ITIL Foundations v3*

Education:

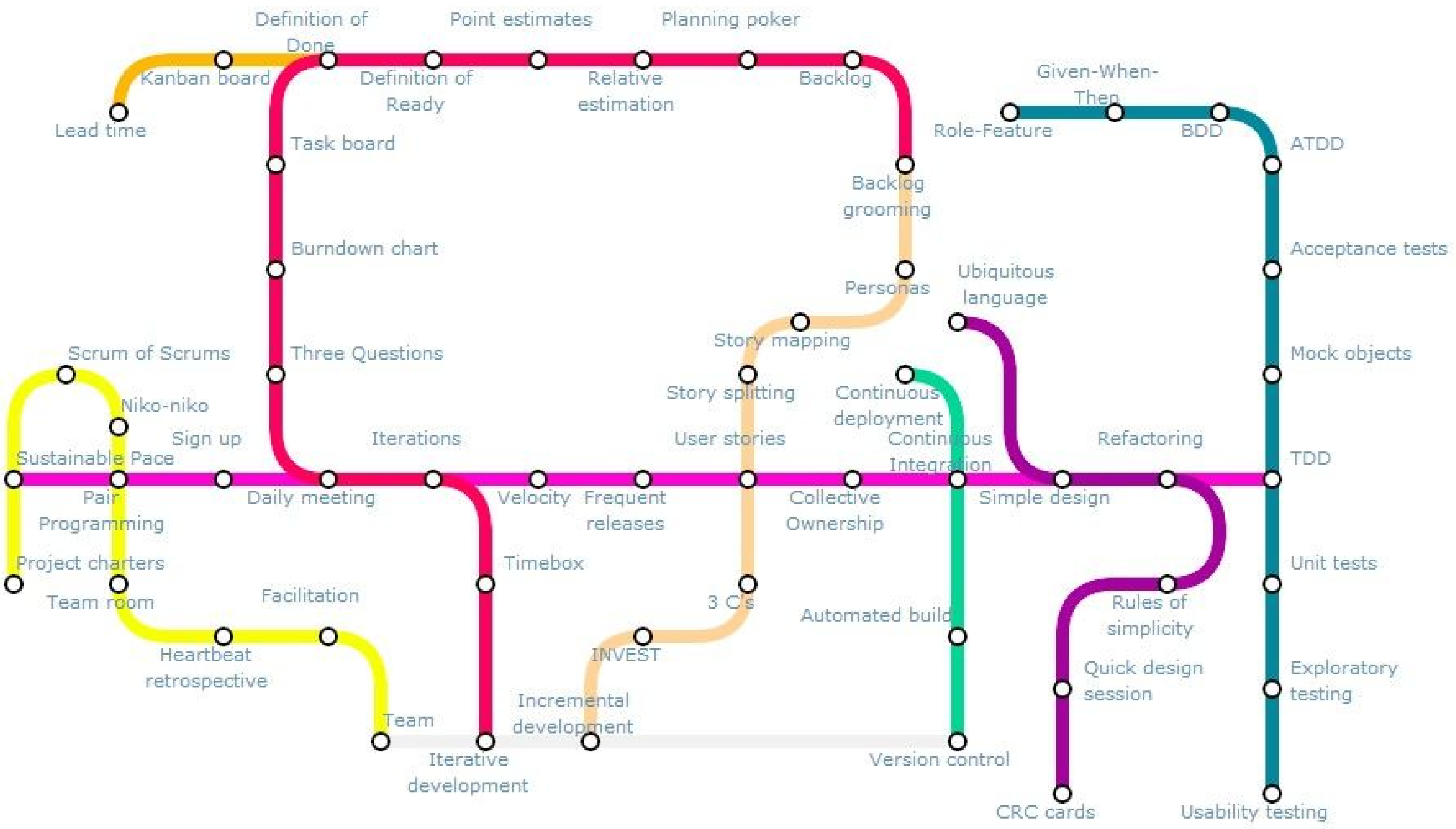
Syracuse University B.S. Management Information Systems
Rensselaer Polytechnic Institute M.S. Software Engineering

Agile

Agile is an umbrella term describing a group of methods and practices in which requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. It promotes adaptive planning, evolutionary development, early delivery, continuous improvement, and encourages rapid and flexible response to change



Agile practices



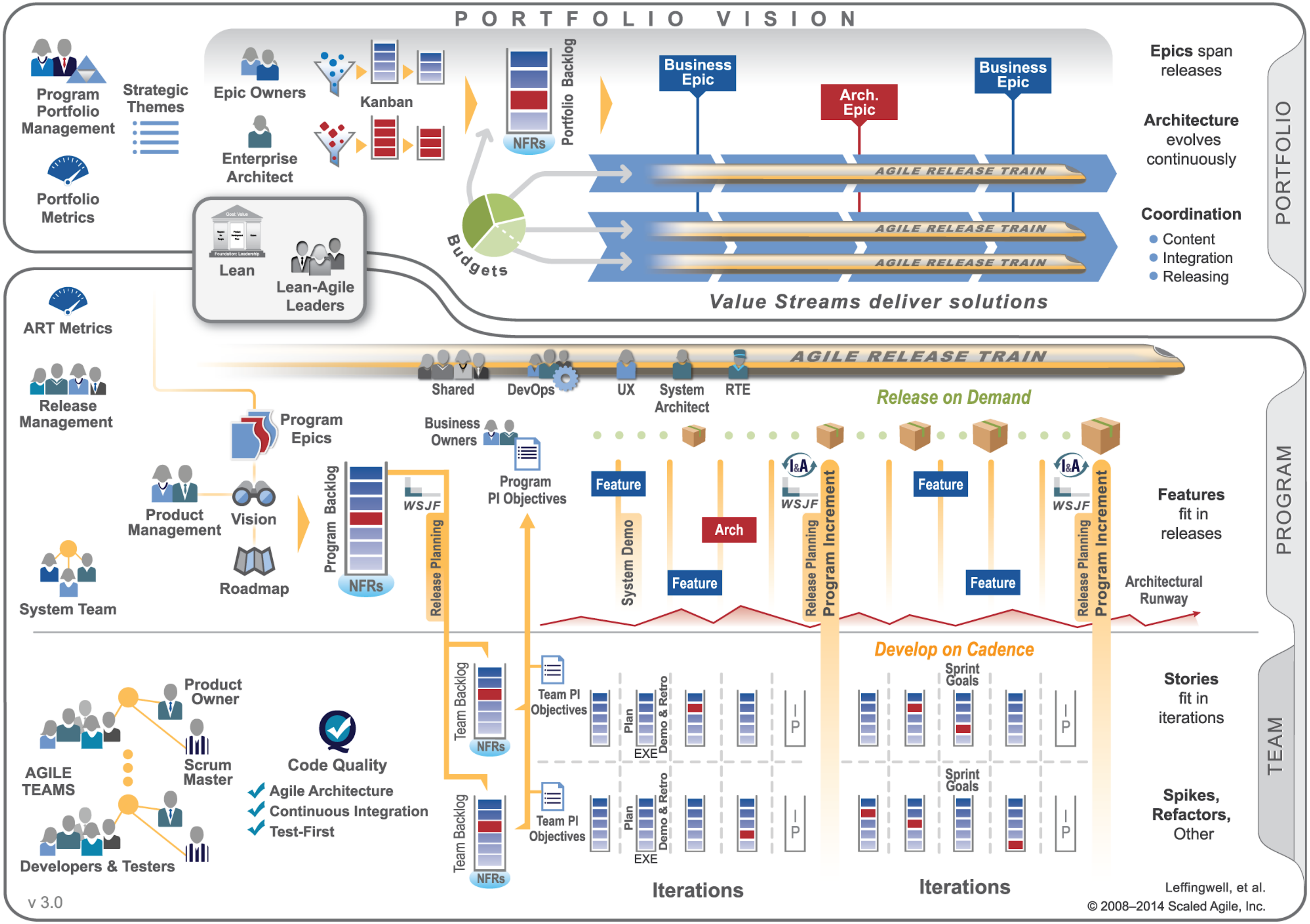
Lines represent practices from the various Agile "tribes" or areas of concern:

- Extreme Programming
- Teams
- Lean
- Scrum
- Product management
- Devops
- Design
- Testing
- Fundamentals

Agile Alliance

Scaled Agile Framework®

SAFe®



PORTFOLIO VISION

PORTFOLIO

Program Portfolio Management

Strategic Themes

Portfolio Metrics

Epic Owners

Enterprise Architect

Kanban

Portfolio Backlog

NFRs

Business Epic

Arch. Epic

Business Epic

AGILE RELEASE TRAIN

AGILE RELEASE TRAIN

AGILE RELEASE TRAIN

Budgets

Value Streams deliver solutions

Lean

Lean-Agile Leaders

Epics span releases

Architecture evolves continuously

Coordination

- Content
- Integration
- Releasing

PROGRAM

ART Metrics

Release Management

System Team

AGILE RELEASE TRAIN

Shared

DevOps

UX

System Architect

RTE

Program Epics

Product Management

Vision

Roadmap

Program Backlog

NFRs

WSJF

Release Planning

Business Owners

Program PI Objectives

Release on Demand

Feature

System Demo

Arch

Feature

I&A

WSJF

Release Planning

Program Increment

Feature

Feature

I&A

WSJF

Release Planning

Program Increment

Features fit in releases

Architectural Runway

TEAM

Develop on Cadence

Product Owner

Scrum Master

AGILE TEAMS

Developers & Testers

Code Quality

- ✓ Agile Architecture
- ✓ Continuous Integration
- ✓ Test-First

Team Backlog

NFRs

Team PI Objectives

Plan

EXE

Demo & Retro

Sprint Goals

IP

Team Backlog

NFRs

Team PI Objectives

Plan

EXE

Demo & Retro

Sprint Goals

IP

Iterations

Iterations

Stories fit in iterations

Spikes, Refactors, Other

v 3.0

Leffingwell, et al.

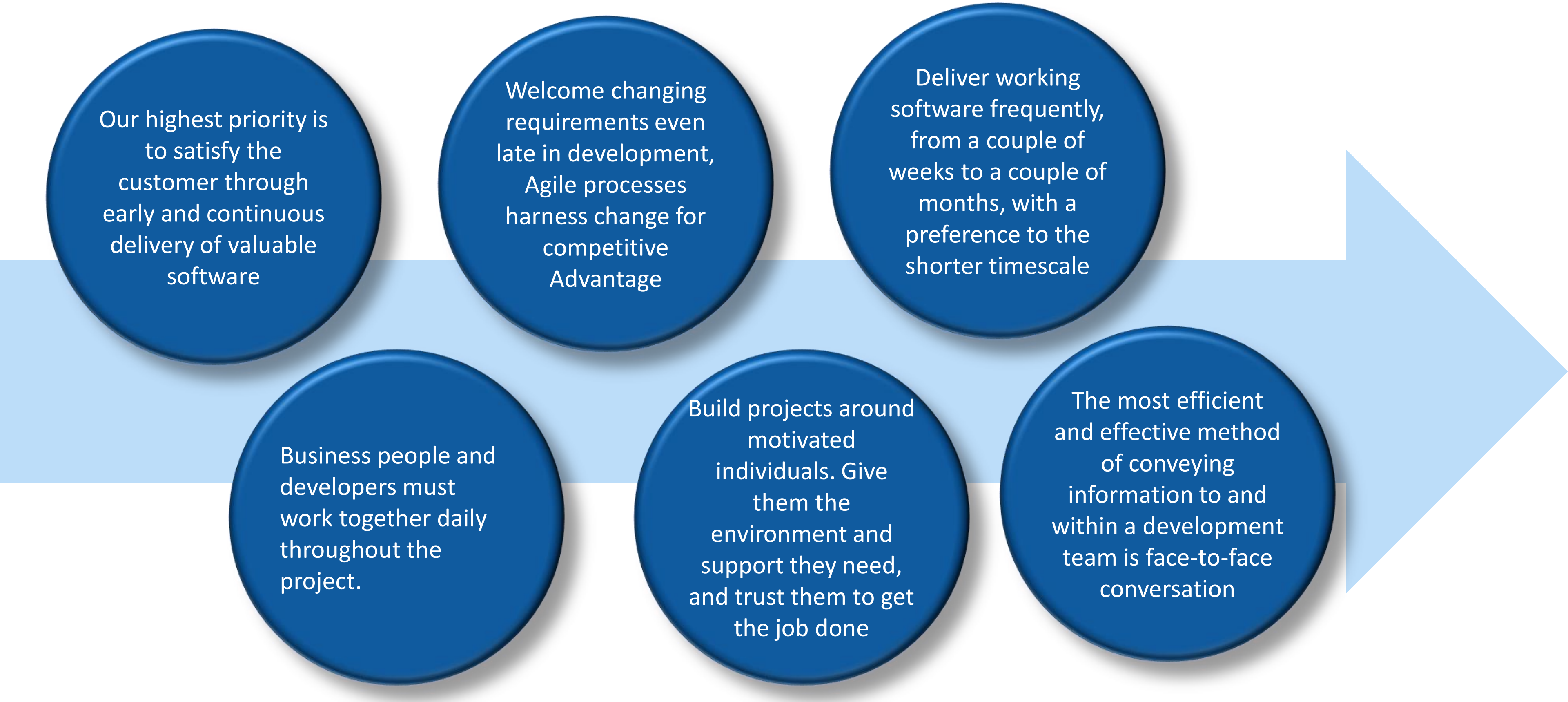
Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions	over	processes and tools
Working software	over	comprehensive documentation
Customer collaboration	over	contract negotiation
Responding to change	over	following a plan

While there is value in the items on the right,
we value the items on the left more.

12 Agile Principles



Our highest priority is to satisfy the customer through early and continuous delivery of valuable software

Welcome changing requirements even late in development, Agile processes harness change for competitive Advantage

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale

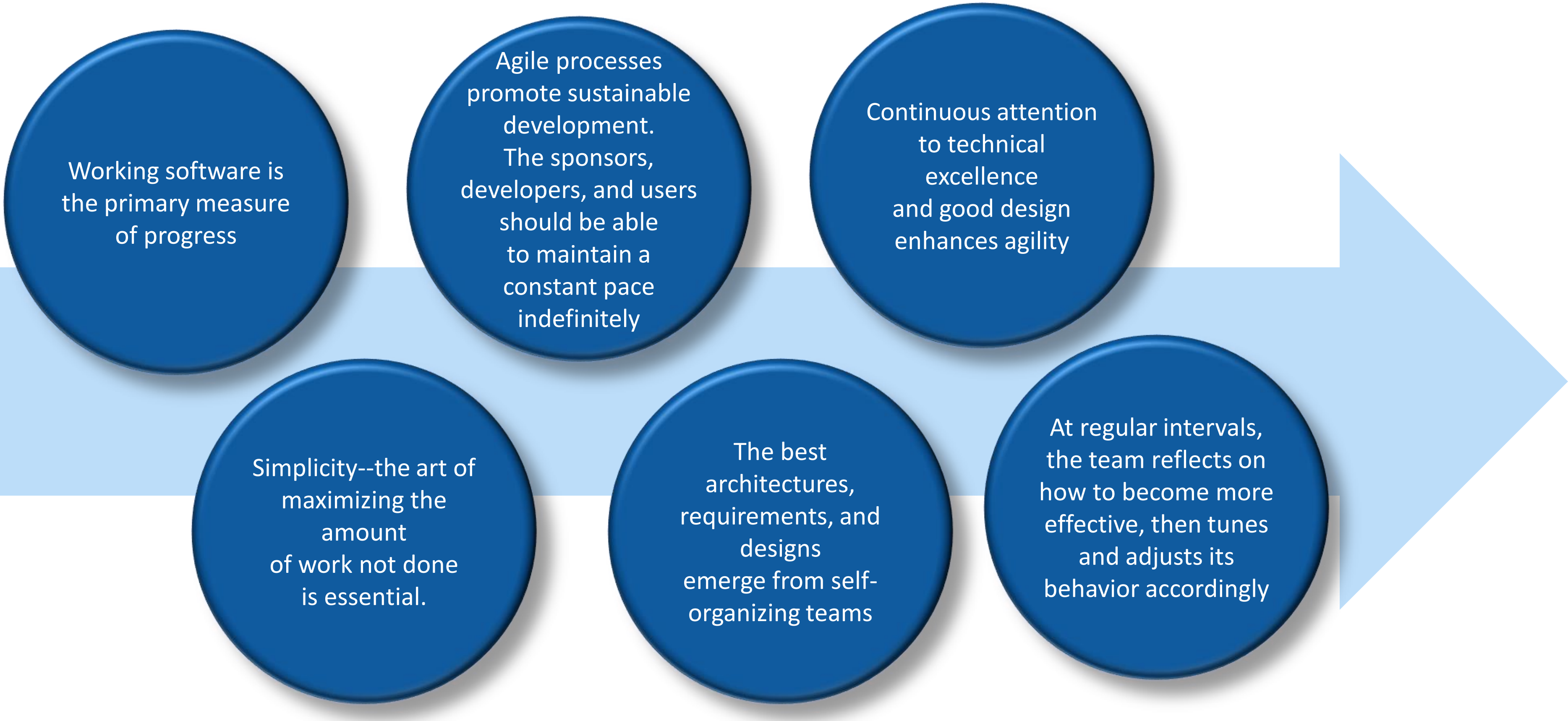
Business people and developers must work together daily throughout the project.

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation

“12 Principles of Agile Development”

12 Agile Principles (continued)



Working software is the primary measure of progress

Agile processes promote sustainable development.
The sponsors, developers, and users should be able to maintain a constant pace indefinitely

Continuous attention to technical excellence and good design enhances agility

Simplicity--the art of maximizing the amount of work not done is essential.

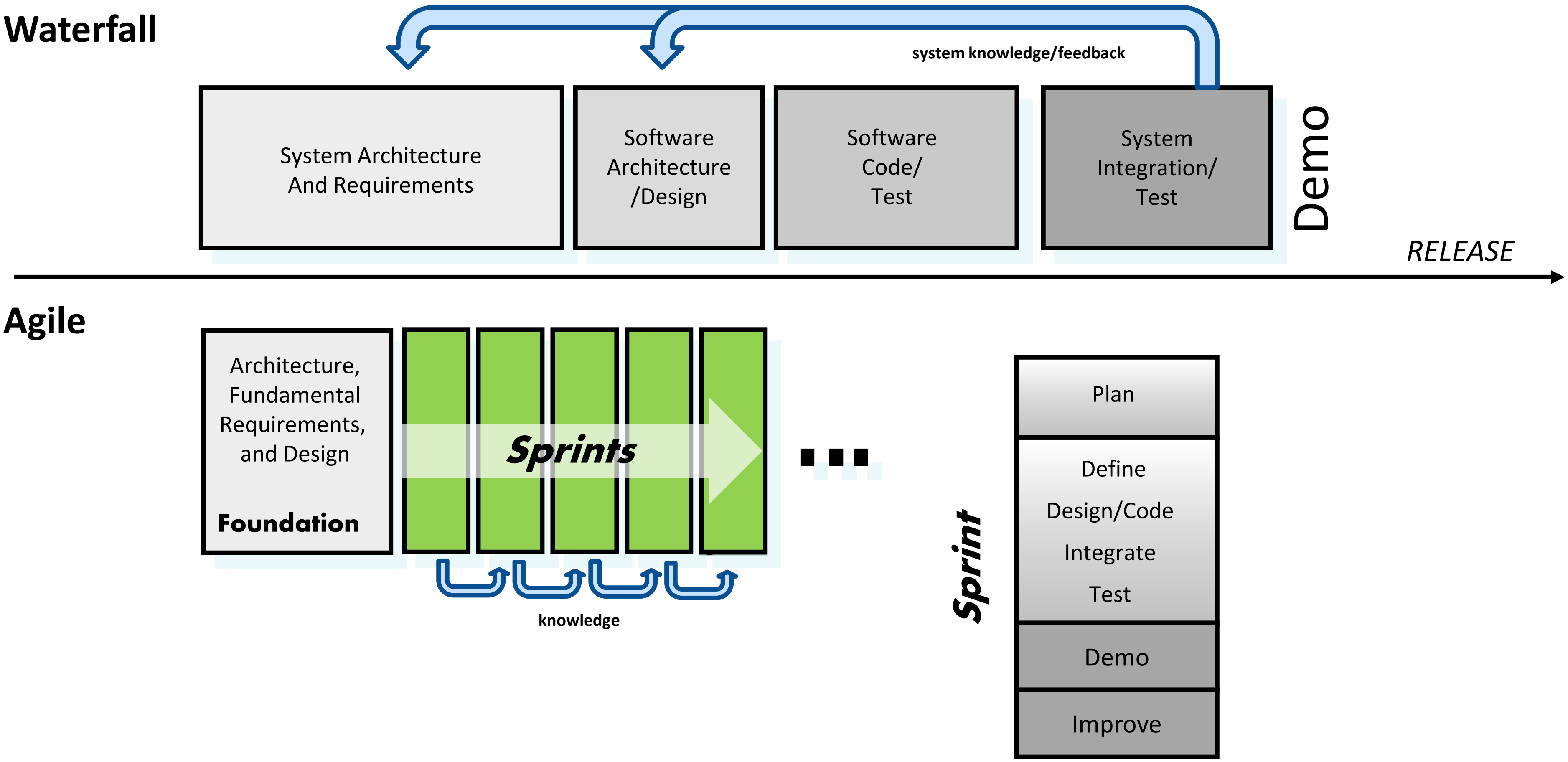
The best architectures, requirements, and designs emerge from self-organizing teams

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly

"12 Principles of Agile Development"

NextGenLM

Agile vs. Waterfall – Development Cycle



Waterfall

system knowledge/feedback

System Architecture And Requirements

Software Architecture /Design

Software Code/ Test

System Integration/ Test

Demo

RELEASE

Agile

Architecture, Fundamental Requirements, and Design

Foundation

Sprints

...

knowledge

Sprint

Plan

Define

Design/Code

Integrate

Test

Demo

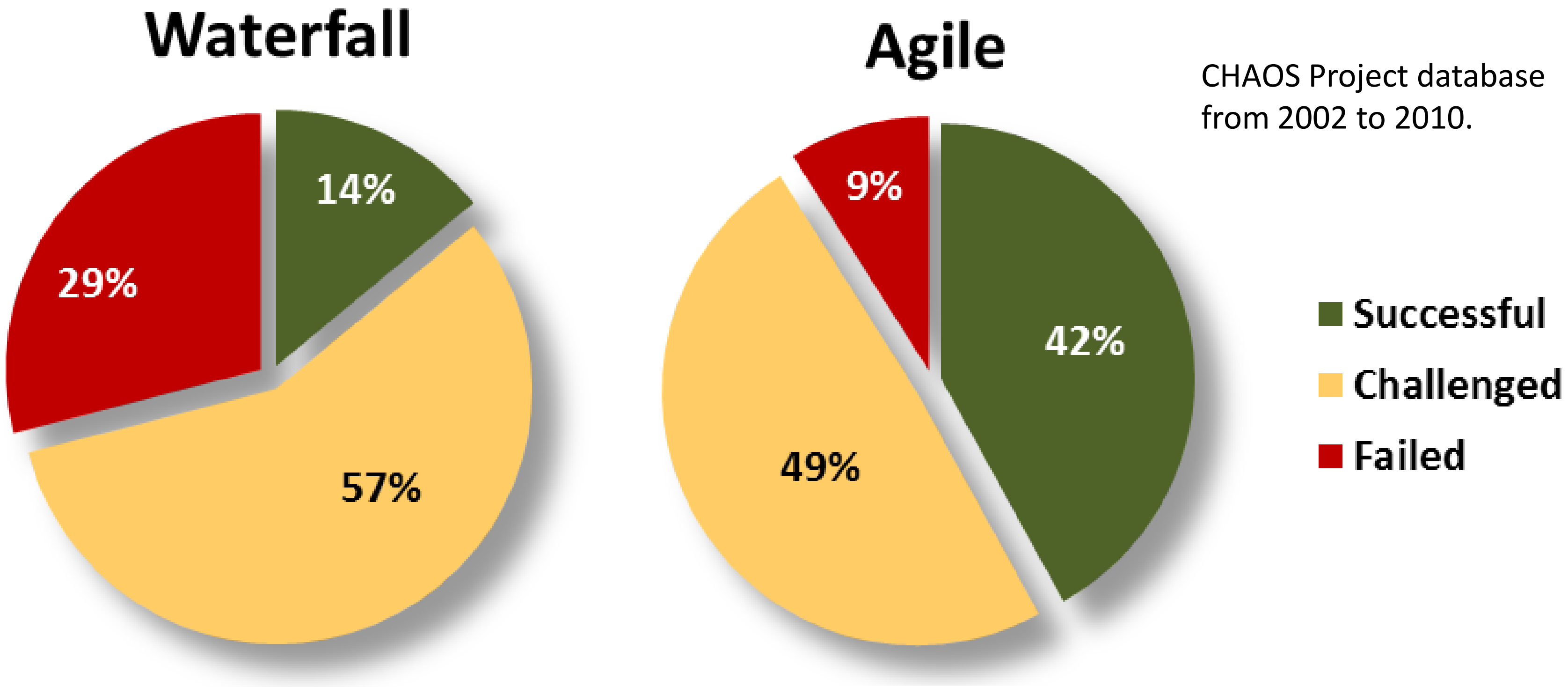
Improve

Agile Features Early Cycles of Development with Feedback

NextGenLM

Agile Program Results

Agile projects are 3X more successful than Waterfall



CHAOS Project database from 2002 to 2010.

Successful = delivered on time, on budget, with all required features and functions
Challenged = Late, over budget, and/or with less than required features and functions
Failed = Cancelled prior to completion or delivered and never used

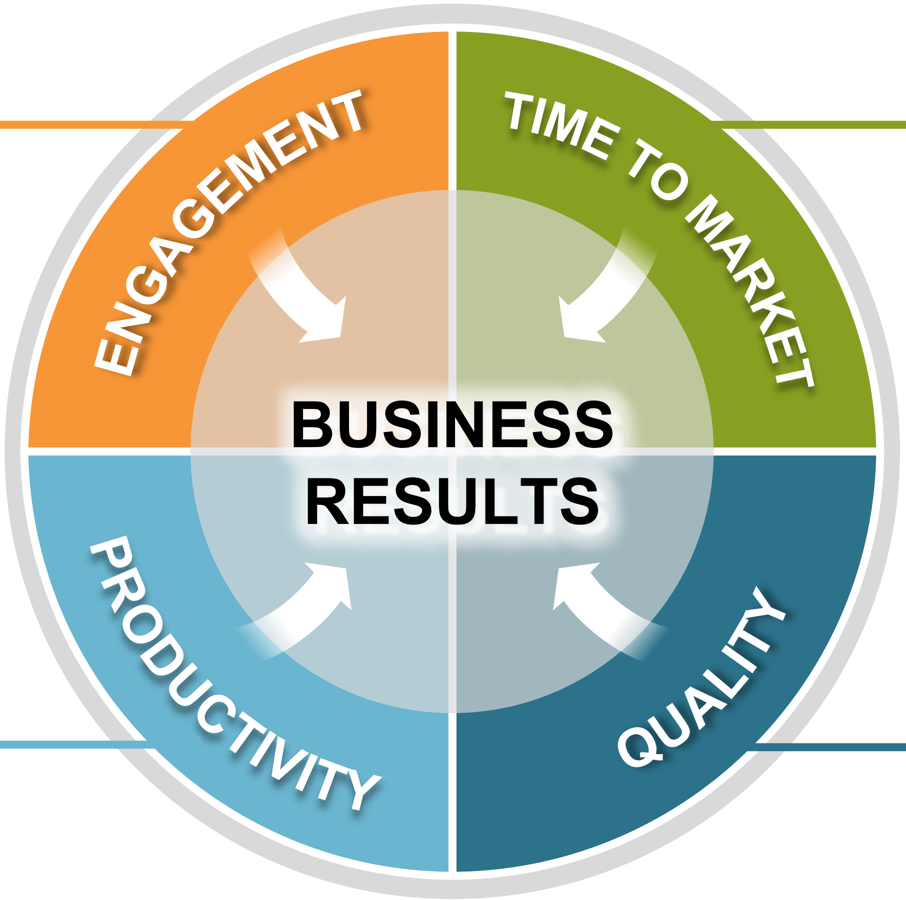
Source: CHAOS Manifesto 2011, The Standish Group International , Inc.

NextGenLM

Why SAFe

Increase in employee engagement

30-75% faster time to market



20-50% increase in productivity

50%+ defect reduction

Leffingwell et al. © 2014 Scaled Agile, Inc.

Results seen at Lockheed Martin

❑ *Increased quality*

❑ *Ability to respond to change*

❑ *Reduced cost by up to 50%*

❑ *Reduced schedules by up to 50%*

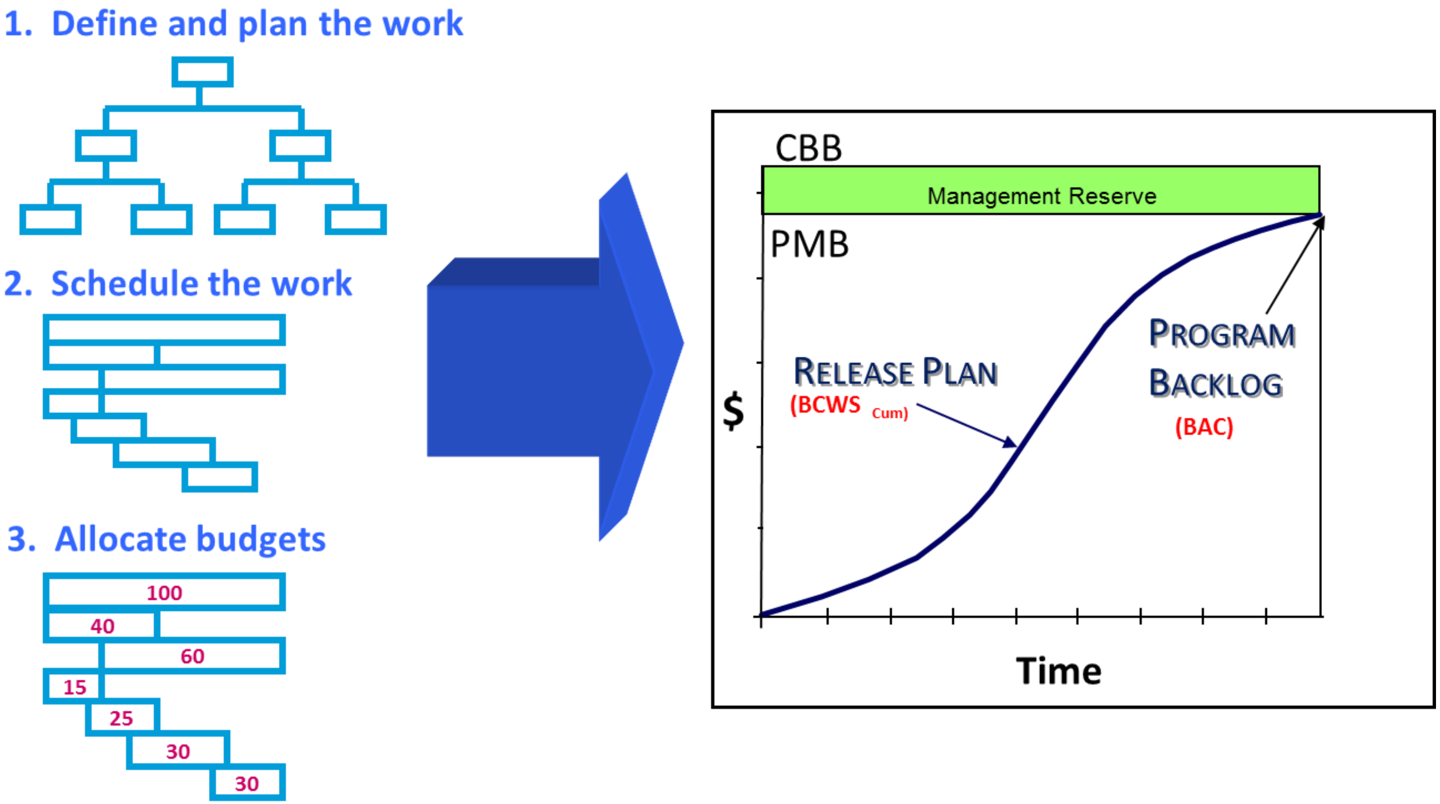
❑ *Reduced defect profiles by 40%*

❑ *Higher morale*

Agile Measurement Baseline (PMB)

Performance Measurement Baseline

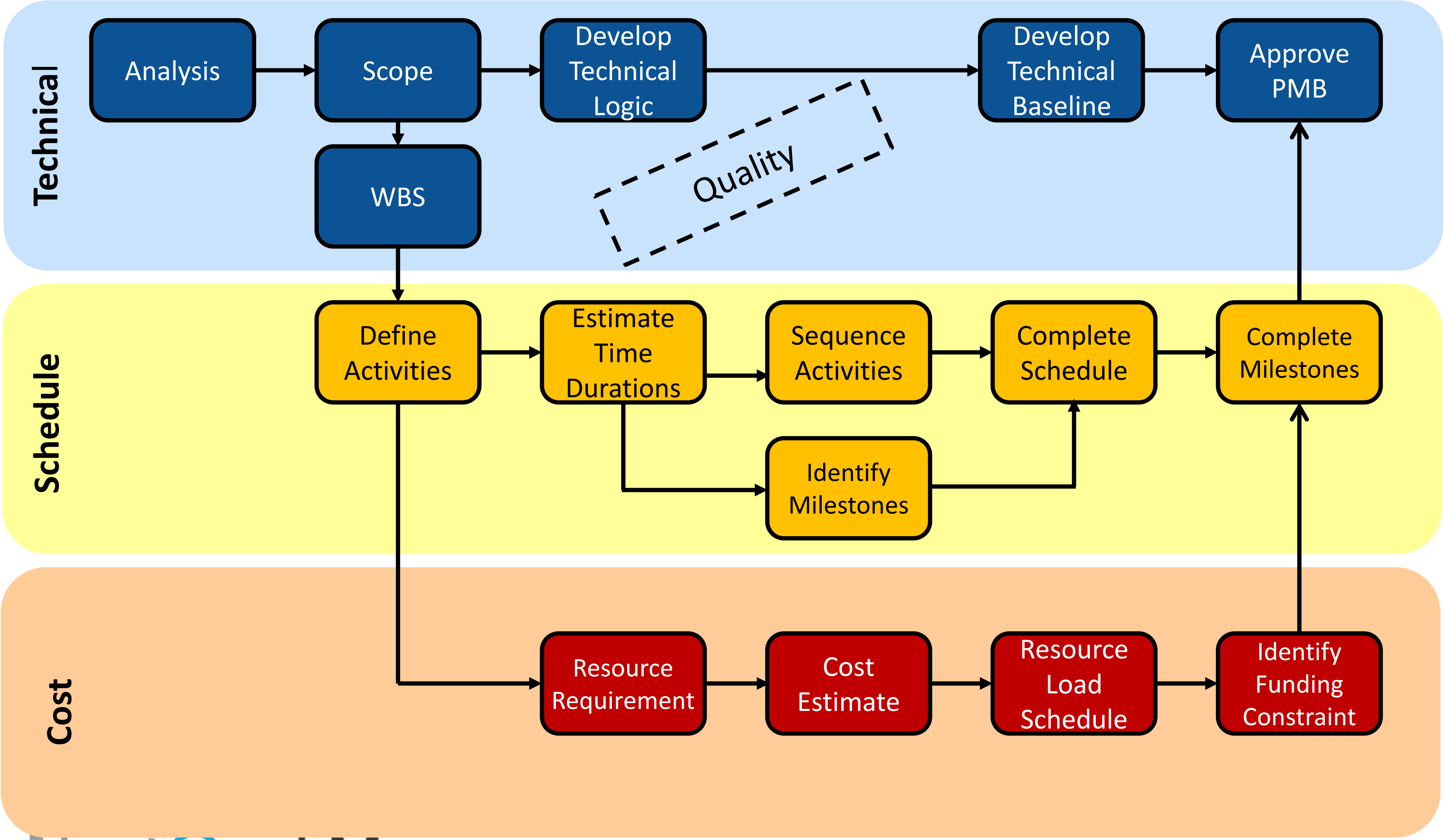
Measuring project performance against a time phased budget plan for accomplishing all work. Performance is measured against scope, schedule, and cost plans.



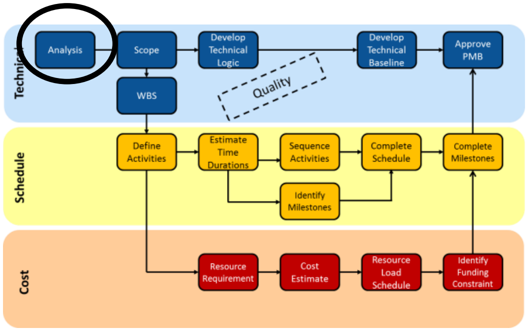
1. Define and plan the work
2. Schedule the work
3. Allocate budgets

NextGenLM

The PMB is actually 3 baselines



Analysis



Identify the Requirements

Requirements Repository

Product Backlog

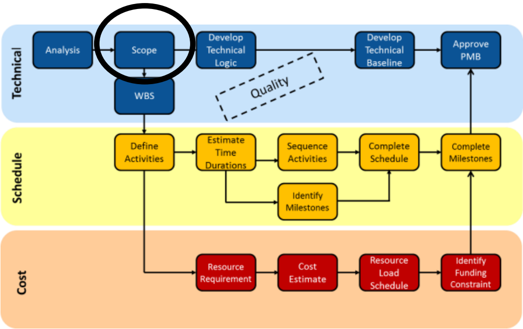
Waterfall	Agile
System Design – A Spec	Epic
Component Design B-Spec	Sub Epic
Software/ Interface Requirements	Features
Detail Requirements	User Stories

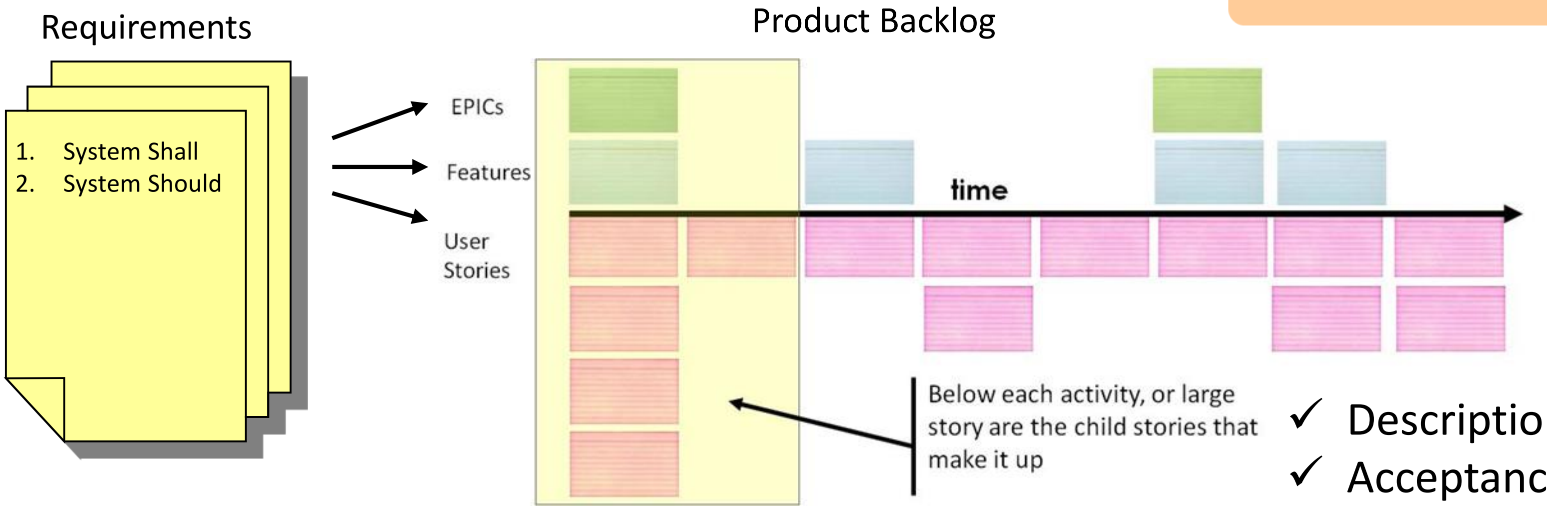
Although we use different terminology we are still gathering and analyzing requirements

NextGenLM

Scope

Based on the analysis of the requirements define the scope of the work. For Agile we place requirements in the form of user stories in a Product backlog.




Requirements

Product Backlog

1. System Shall
2. System Should

EPICs

Features

User Stories

time

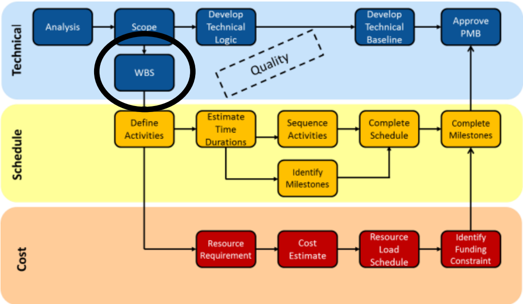
Below each activity, or large story are the child stories that make it up

- ✓ Description
- ✓ Acceptance Criteria
- ✓ Deliverables
- ✓ Constraints

Link your requirements to the product backlog which is where the scope of work is defined.

Work Breakdown Structure (WBS)

Agile programs utilize a release or capability centric work break down structure that focus on business outcomes as opposed to functional based work break down structures, that place the emphasis on inputs such as software, systems, test, etc..

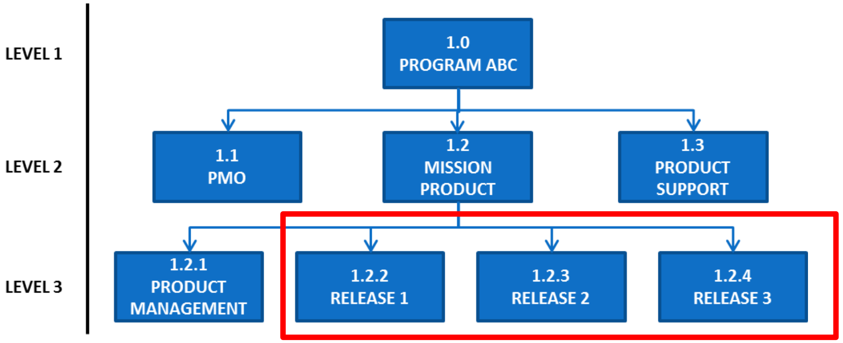


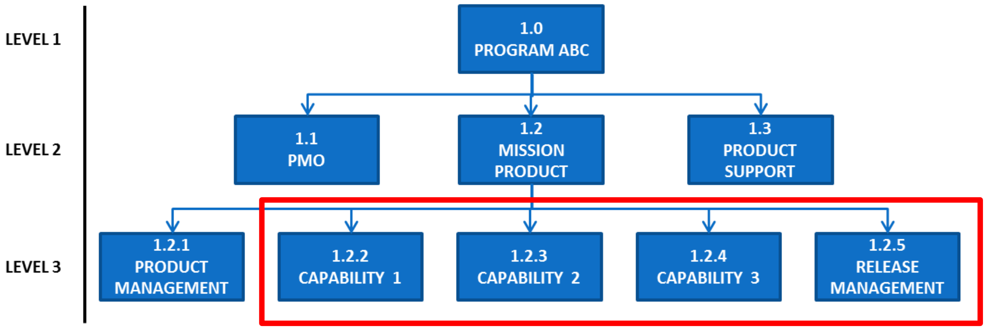
RELEASE CENTRIC

The customer views the product in terms of release. An example of this might be a large satellite ground system where the releases are based around major system events such as launch support, initial calibration, initial operations, and full system operations.

CAPABILITY CENTRIC

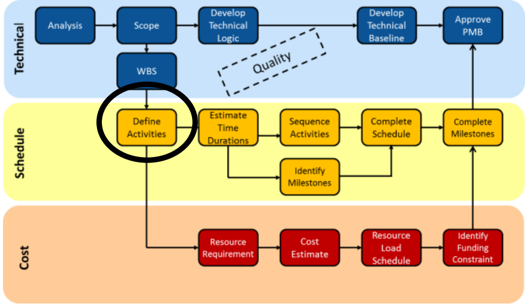
The customer views the product in terms of a set of discrete capabilities, where the releases are primarily viewed as time boxes for the ongoing and sustained delivery of Features. The release content may change greatly over time based upon changing priorities





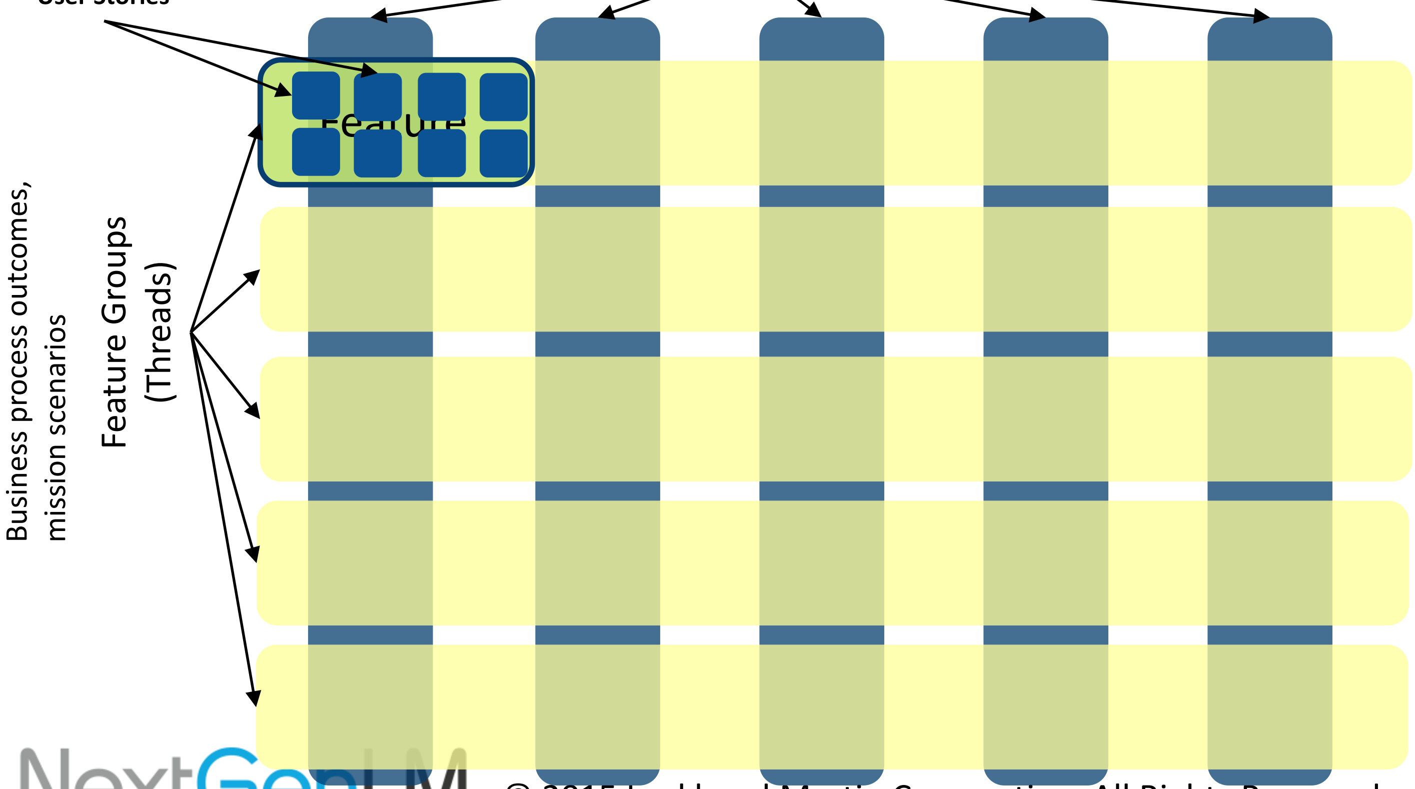
Define Activities

Define Activities for backlog, start with Epics and Feature groups and iteratively decompose into features and user stories



Epics

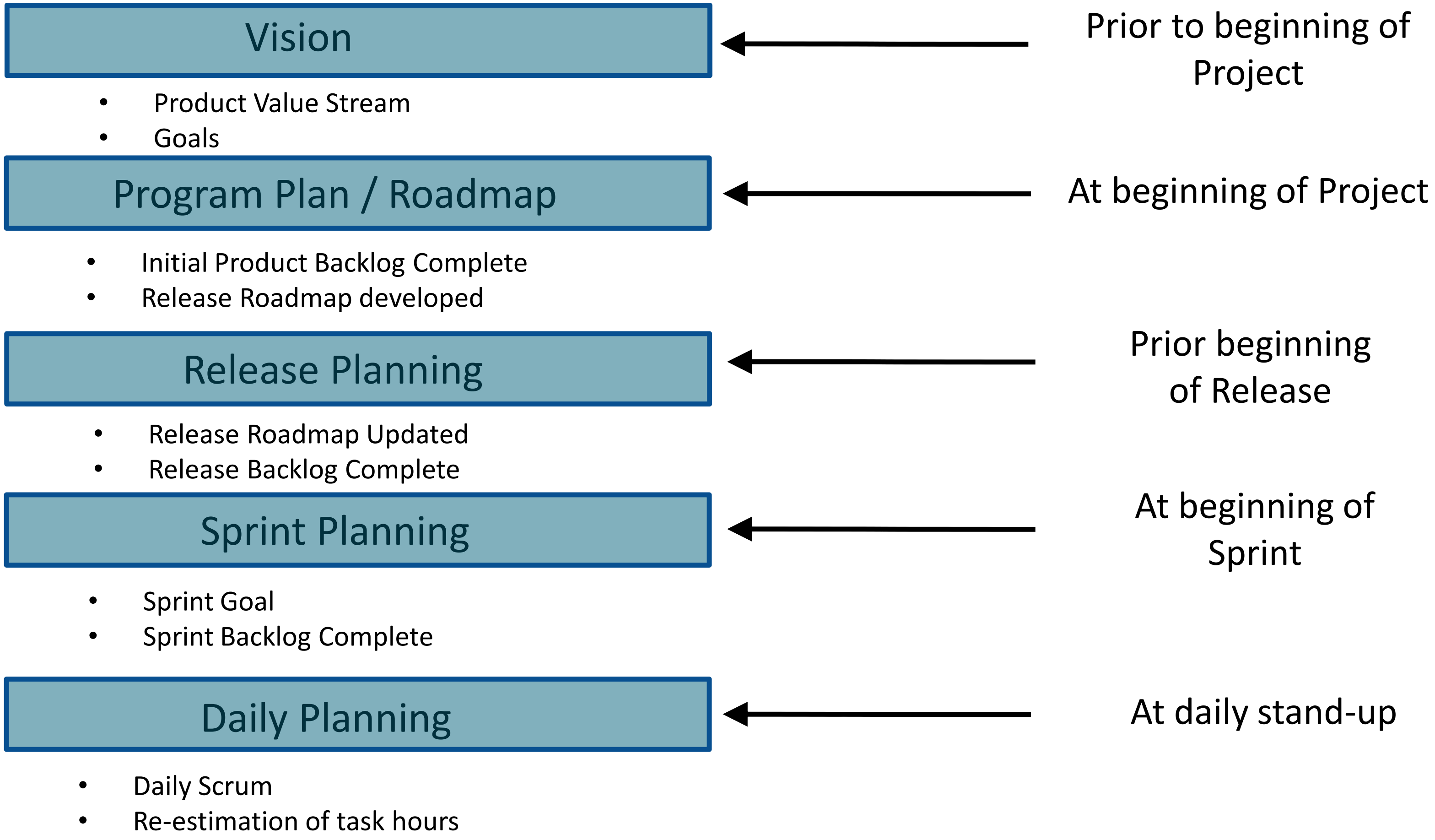
Cross cutting system capabilities, persistent through life-cycle



Business process outcomes, mission scenarios

NextGenLM © 2015 Lockheed Martin Corporation. All Rights Reserved

Five Levels of Planning



Vision ← Prior to beginning of Project

- Product Value Stream
- Goals

Program Plan / Roadmap ← At beginning of Project

- Initial Product Backlog Complete
- Release Roadmap developed

Release Planning ← Prior beginning of Release

- Release Roadmap Updated
- Release Backlog Complete

Sprint Planning ← At beginning of Sprint

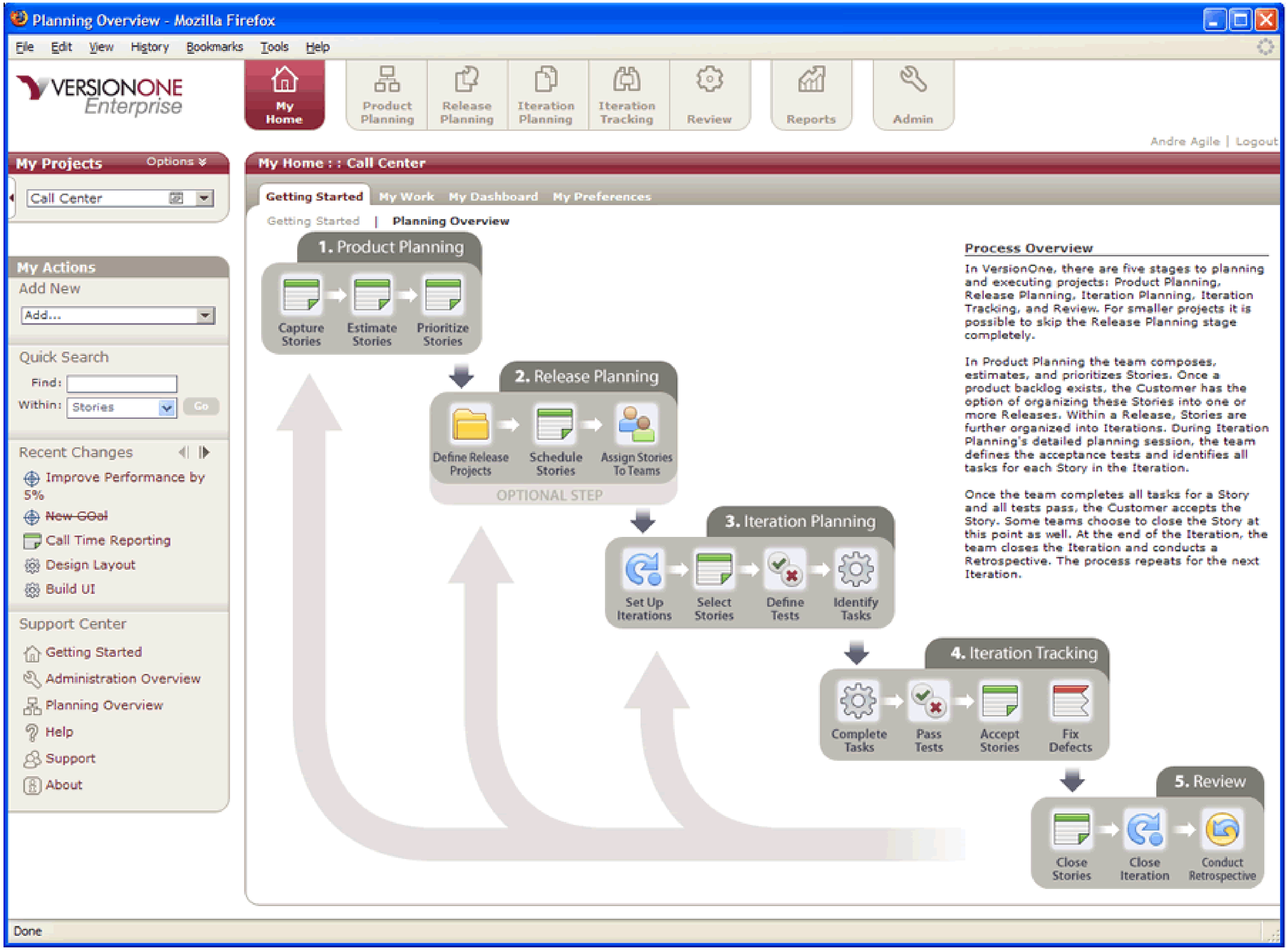
- Sprint Goal
- Sprint Backlog Complete

Daily Planning ← At daily stand-up

- Daily Scrum
- Re-estimation of task hours

NextGenLM

VersionOne



Process Overview

In VersionOne, there are five stages to planning and executing projects: Product Planning, Release Planning, Iteration Planning, Iteration Tracking, and Review. For smaller projects it is possible to skip the Release Planning stage completely.

In Product Planning the team composes, estimates, and prioritizes Stories. Once a product backlog exists, the Customer has the option of organizing these Stories into one or more Releases. Within a Release, Stories are further organized into Iterations. During Iteration Planning's detailed planning session, the team defines the acceptance tests and identifies all tasks for each Story in the Iteration.

Once the team completes all tasks for a Story and all tests pass, the Customer accepts the Story. Some teams choose to close the Story at this point as well. At the end of the Iteration, the team closes the Iteration and conducts a Retrospective. The process repeats for the next Iteration.

Hierarchy

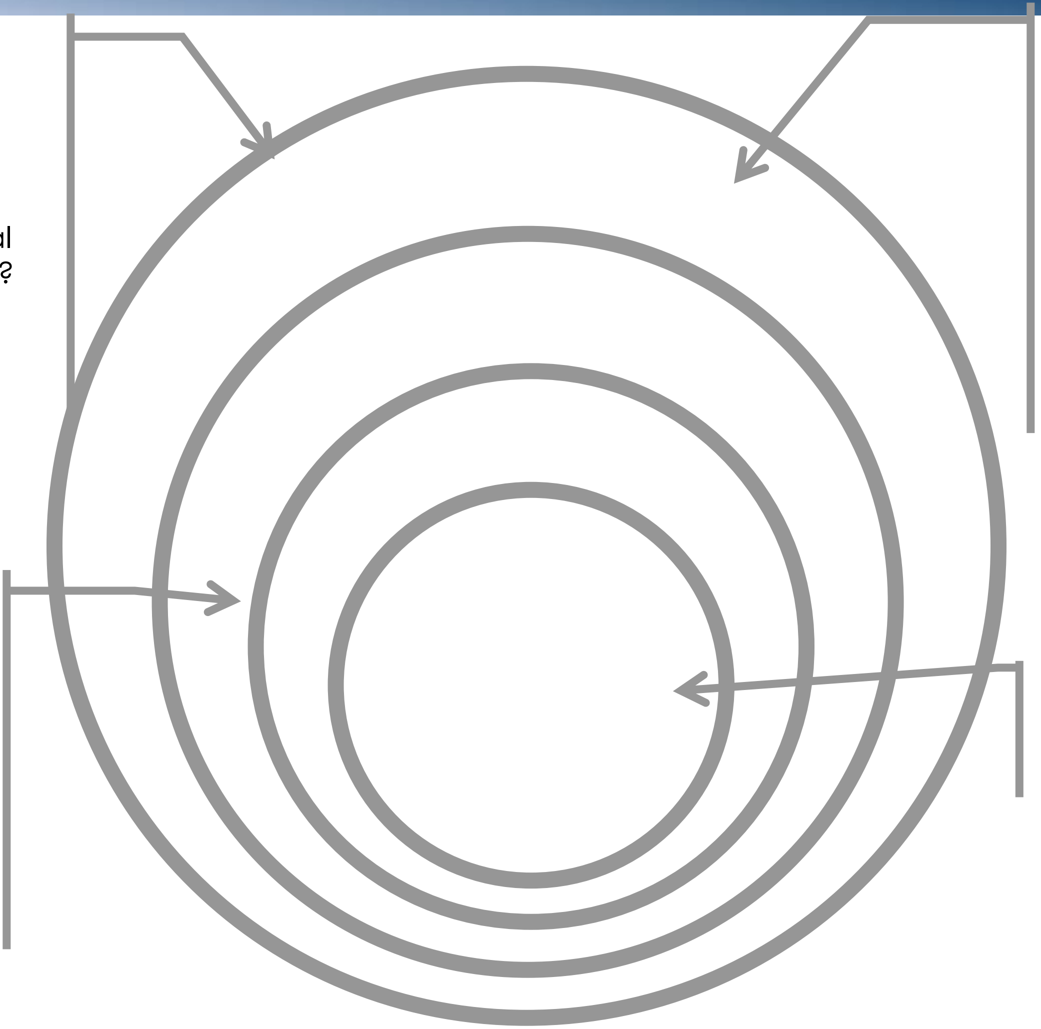
- Epic – May span multiple releases, large capability
- Feature – Completed within Release, business process based
- User Story – Completed within Sprint
- Tasks – 2 to 8 hours

Decomposition

<table>
<tr><td colspan="8">Epic</td><td colspan="8">Epic</td></tr>
<tr><td colspan="4">Feature</td><td colspan="4">Feature</td><td colspan="4">Feature</td><td colspan="4">Feature</td></tr>
<tr><td colspan="2">User Story</td><td colspan="2">User Story</td><td colspan="2">User Story</td><td colspan="2">User Story</td><td colspan="2">User Story</td><td colspan="2">User Story</td><td colspan="2">User Story</td><td colspan="2">User Story</td></tr>
<tr><td>Task</td><td>Task</td><td>Task</td><td>Task</td><td>Task</td><td>Task</td><td>Task</td><td>Task</td><td>Task</td><td>Task</td><td>Task</td><td>Task</td><td>Task</td><td>Task</td><td>Task</td><td>Task</td></tr>
</table>

Agile breaks work down into inch stones, above is the hierarchy

The product owner plans the product in layers



Product
or **Program**
Backlog

What business / technical objectives will be fulfilled?

(Business & Architectural Epics)

Release Backlog

Incremental value (portions of Epics)

Subset of Business and Technical Objectives

Release Roadmap

(Features)

Iteration / Sprint Backlog

Incremental Value

Smallest subset of Business and Technical objectives

(User Stories)

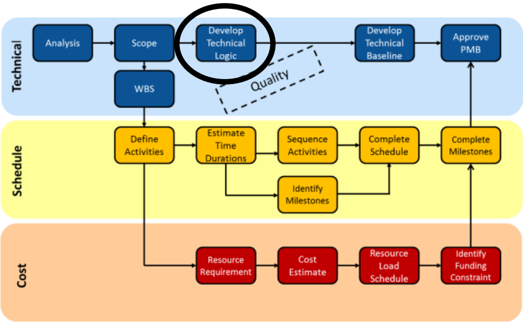
Daily Backlog

Tasks to complete stories

Define technical logic

- ✓ How much architectural runway
- ✓ Incremental pattern utilized
- ✓ Artifacts required
- ✓ Identify systems we interface
- ✓ Non-functional requirements
- ✓ Accreditation requirements

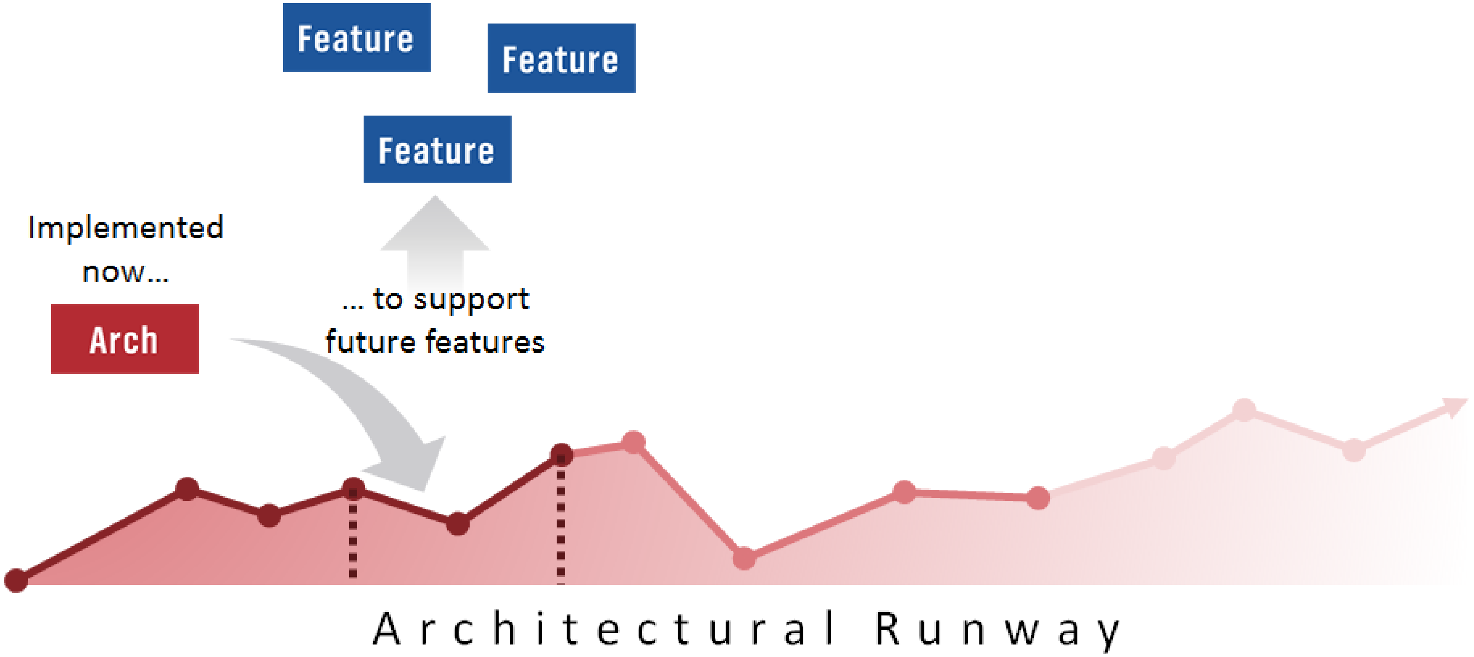
In Agile we keep this at a high level



Technical
Analysis
Scope
Develop Technical Logic
Develop Technical Baseline
Approve PMB
WBS
Quality

Schedule
Define Activities
Estimate Time Durations
Sequence Activities
Complete Schedule
Complete Milestones
Identify Milestones

Cost
Resource Requirement
Cost Estimate
Resource Load Schedule
Identify Funding Constraint



Feature

Feature

Feature

Implemented now…

Arch

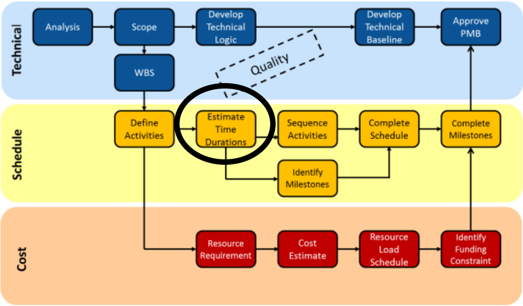
… to support future features

Architectural Runway

NextGenLM

Estimate time durations

Size and duration estimates can be developed using any combination of the 4 methods below. In Agile we will estimate capabilities (Epic/Features) vs Functions (Software / Test)



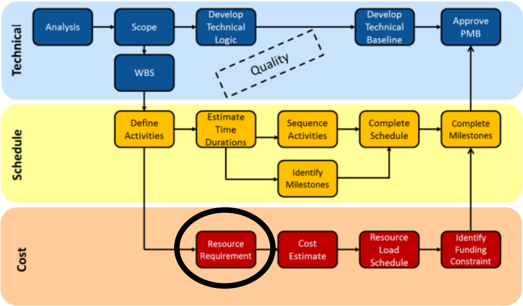
Story point estimation combines expert judgment

With analogous estimating

Method	Description	Pro	Con
Expert Judgment	Judgment guided by subject matter experts based on historical experience	Rapid estimates based on a position of knowledge	Could miss variables and be too heavily weighted on single opinion
Analogous Estimating	Estimate parameters of project based on duration, budget, size, weight complexity. Adjusting for differences	Estimates proven on another project of similar size and complexity	Dependent on having projects of similar size and complexity
Parametric Modeling	Estimates performed based on variables such as function points or SLOC using SEER-SEM or Cocomo.	Provides and objective metric based on historical analysis of similar projects	With the 3rd and 4th generation languages, SLOC becomes less meaningful
3 point Estimates	Estimates based on a weighted average of most likely, optimistic, and pessimistic estimates	Looks at multiple points of view, and considers uncertainty and risk	Difficult to estimate large projects with.

Determine resource requirements

Determine the staffing profile of your project, based on skills sets required. The difference with Agile is that we are going to estimate the team requirements as opposed to individual



Projected	Skill	Level
Tom A	Scrum Master/ Software	5
Robin D	Software Developer	4
Ian B	Software Developer	3
Scott Y	Software Developer	2
Jeff T	Requirements Analyst	3
Helen W	Test Engineer	4
Paul R	Test Engineer	3
James B	Database Engineer	4
		3.5

Develop and average labor cost across team

Hint:
Who you need is not necessarily who you have today

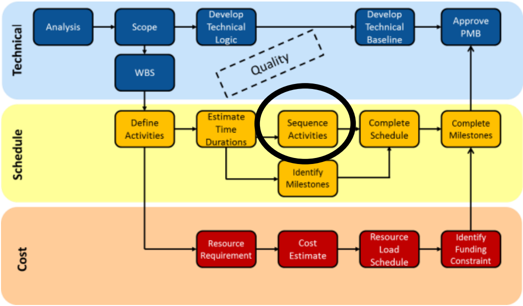
Hint:
The best team results in a ***3*** *to* ***3.5*** *when levels averaged across resources*

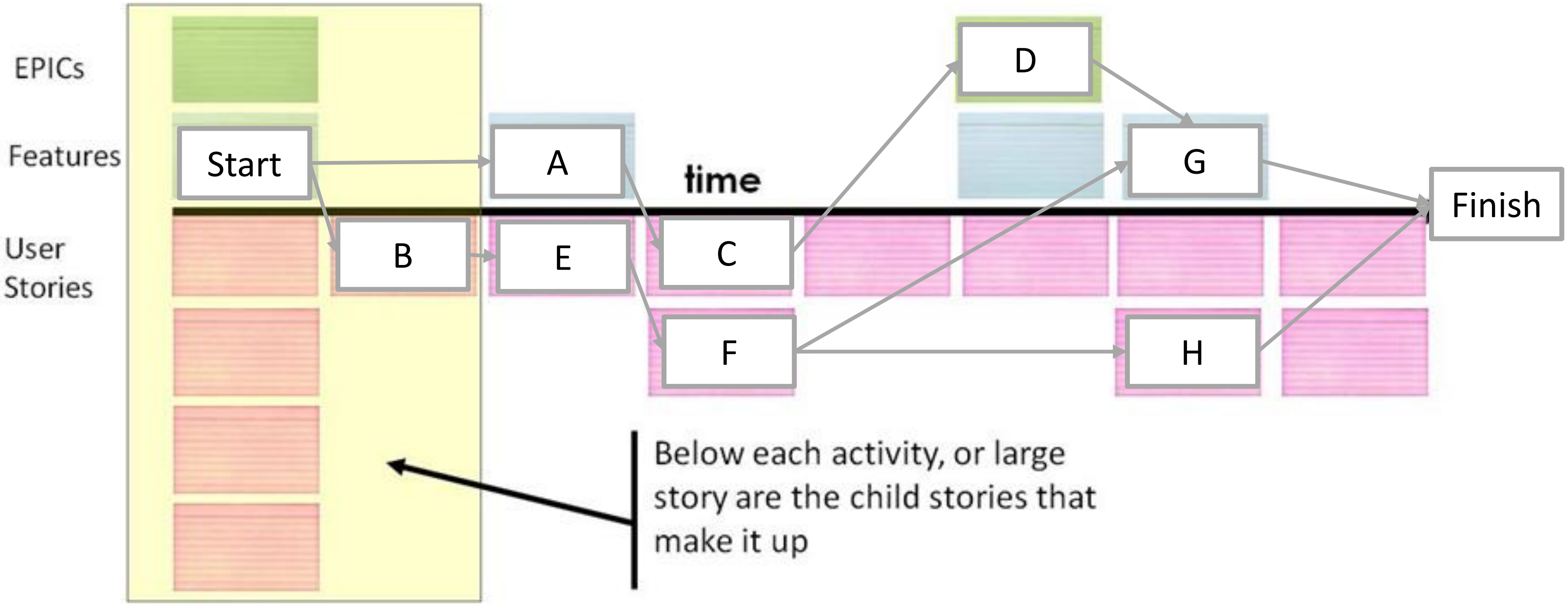
Right size teams, higher levels don't always mean higher productivity

NextGenLM

Sequence activities

Program will sequence their activities. In Agile programs we refer to this as story mapping. On traditional projects sequencing activities is known as the Precedence Diagramming Method (PDM).



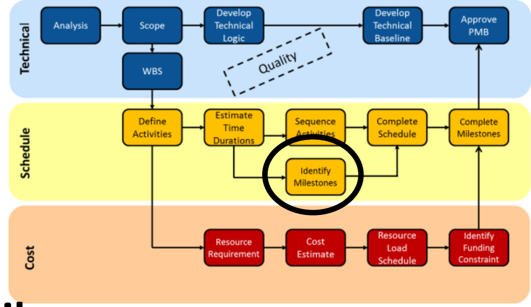


Below each activity, or large story are the child stories that make it up

NextGenLM

Identify milestones

Identify and list their key milestones. However in Agile programs we focus on outcomes as opposed to document and design reviews to take credit.



Traditional

Milestone	Date
Initial Baseline Review (IBR)	3/4/2014
System Requirements Review (SRR)	5/27/2014
Preliminary Design Review (PDR)	8/19/2014
Critical Design Review (CDR)	11/11/2014
Test Readiness Review (TRR)	2/3/2015
Operational Readiness Review (ORR)	4/28/2015
Project Closeout Review (PCR)	7/21/2015

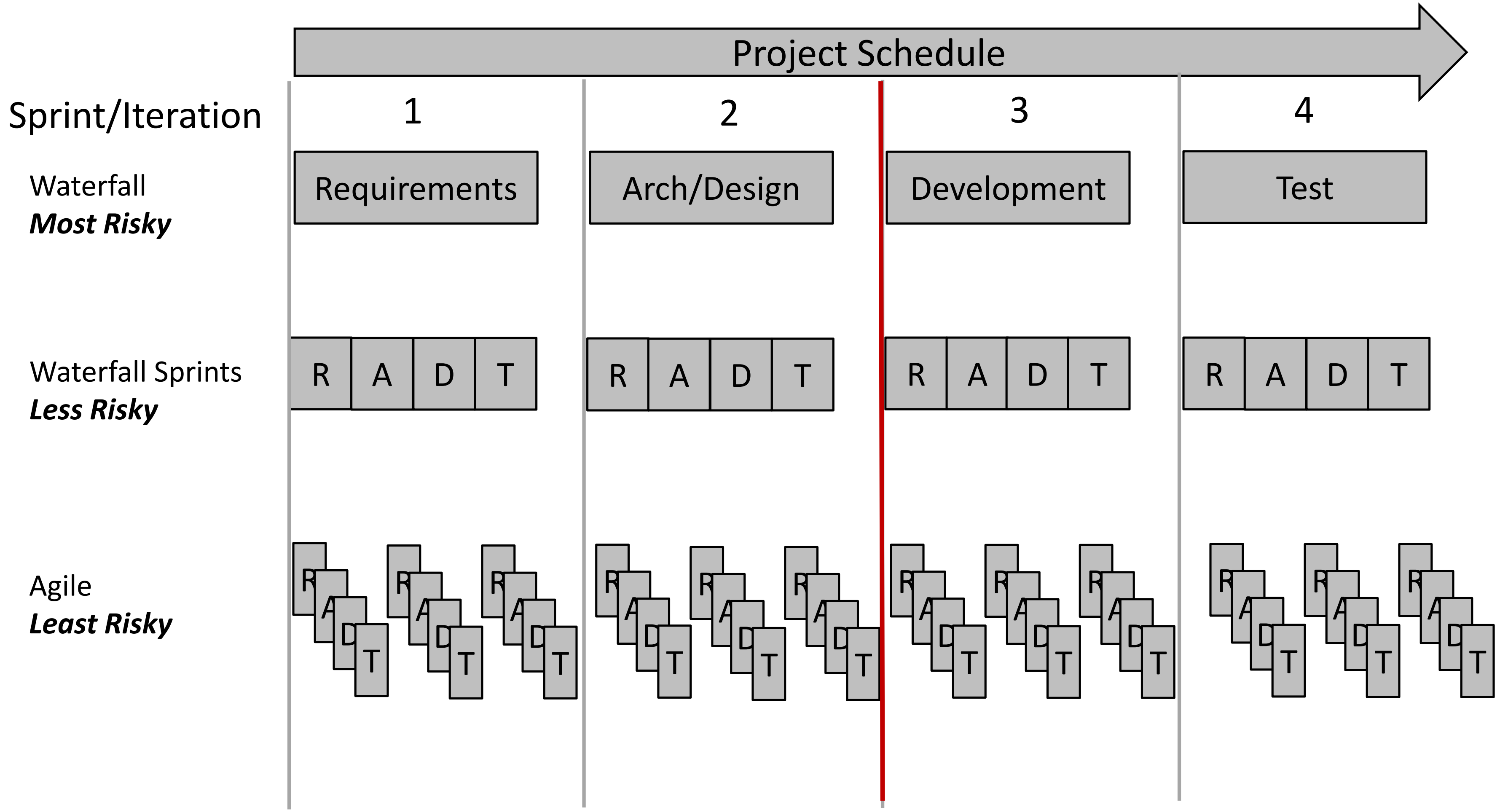
Agile

Milestone	Date
Product Vision Complete	3/4/2014
Product Plan / Roadmap Complete	3/18/2014
Initial Baseline Review (IBR)	4/15/2014
Release Roadmap complete	5/27/2014
Release 1 Demo (Feature 1-4 complete)	8/19/2014
Release 2 Demo (Feature 5-9 Complete)	11/11/2014
Release 3 Demo (Feature 10-13 Complete)	2/3/2015
Release N Demo (Feature 14-17 Complete)	4/28/2015
Project Closeout Review (PCR)	7/21/2015

What do I have on 02/03/2015 on my project?

NextGenLM

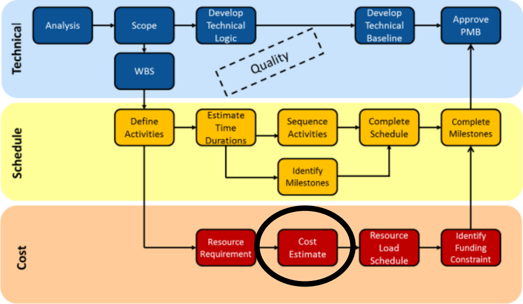
Agile uses time boxing to localize risk



NextGenLM

Cost estimate

There is very little difference in how teams estimate costs between Agile and traditional projects. We often see many efficiencies and risk reductions which enable Agile projects to be lower cost, when run properly.



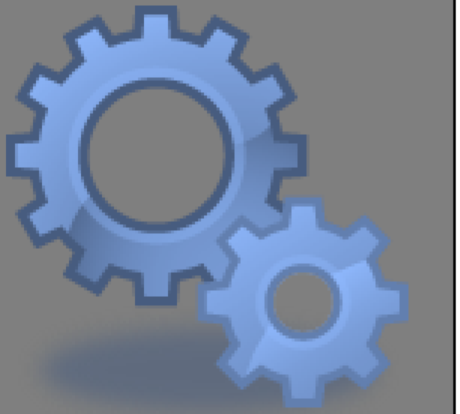
Scope
(Epic/ Feature)

Schedule

Resource Plan

Risks

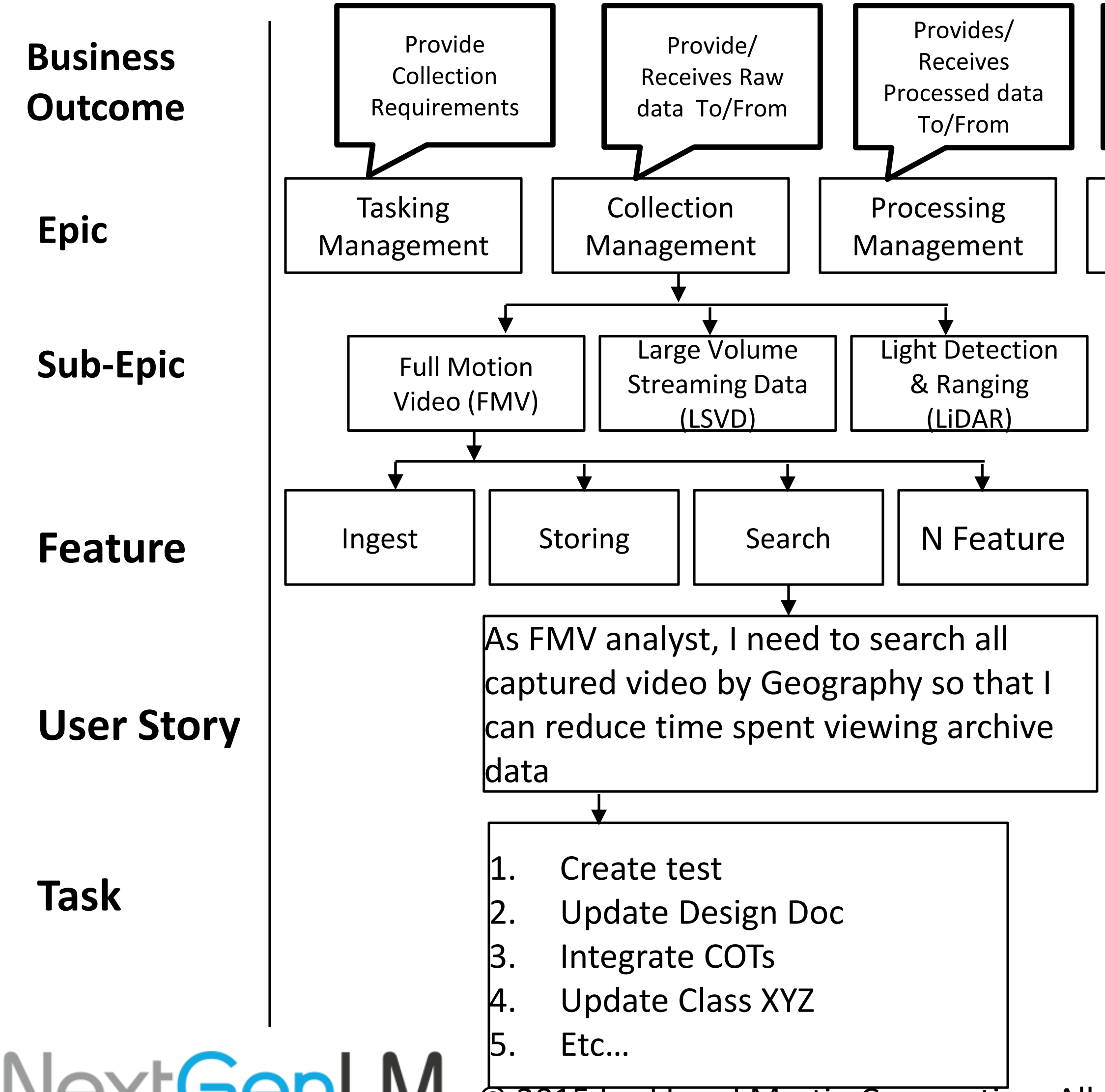
Cost Estimation



Project estimates are a range that will continually need to be reviewed and refined

NextGenLM

Develop technical baseline



Business Outcome

Epic

Sub-Epic

Feature

User Story

Task

Provide Collection Requirements

Provide/ Receives Raw data To/From

Provides/ Receives Processed data To/From

Provides/ Receives Finished data To/From

Intelligence Information , Guidance, support

Tasking Management

Collection Management

Processing Management

Exploitation Management

Dissemination Management

Full Motion Video (FMV)

Large Volume Streaming Data (LSVD)

Light Detection & Ranging (LiDAR)

Ingest

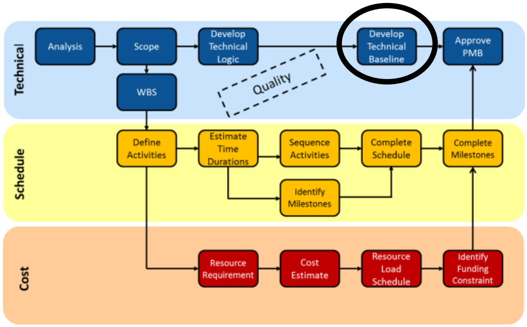
Storing

Search

N Feature

As FMV analyst, I need to search all captured video by Geography so that I can reduce time spent viewing archive data

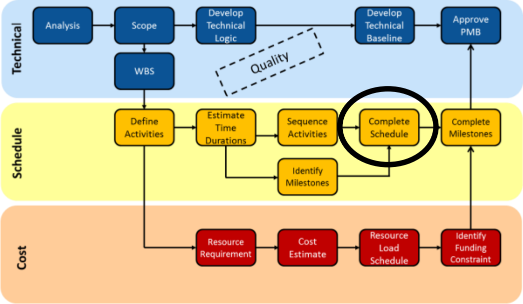
1. Create test
2. Update Design Doc
3. Integrate COTs
4. Update Class XYZ
5. Etc…

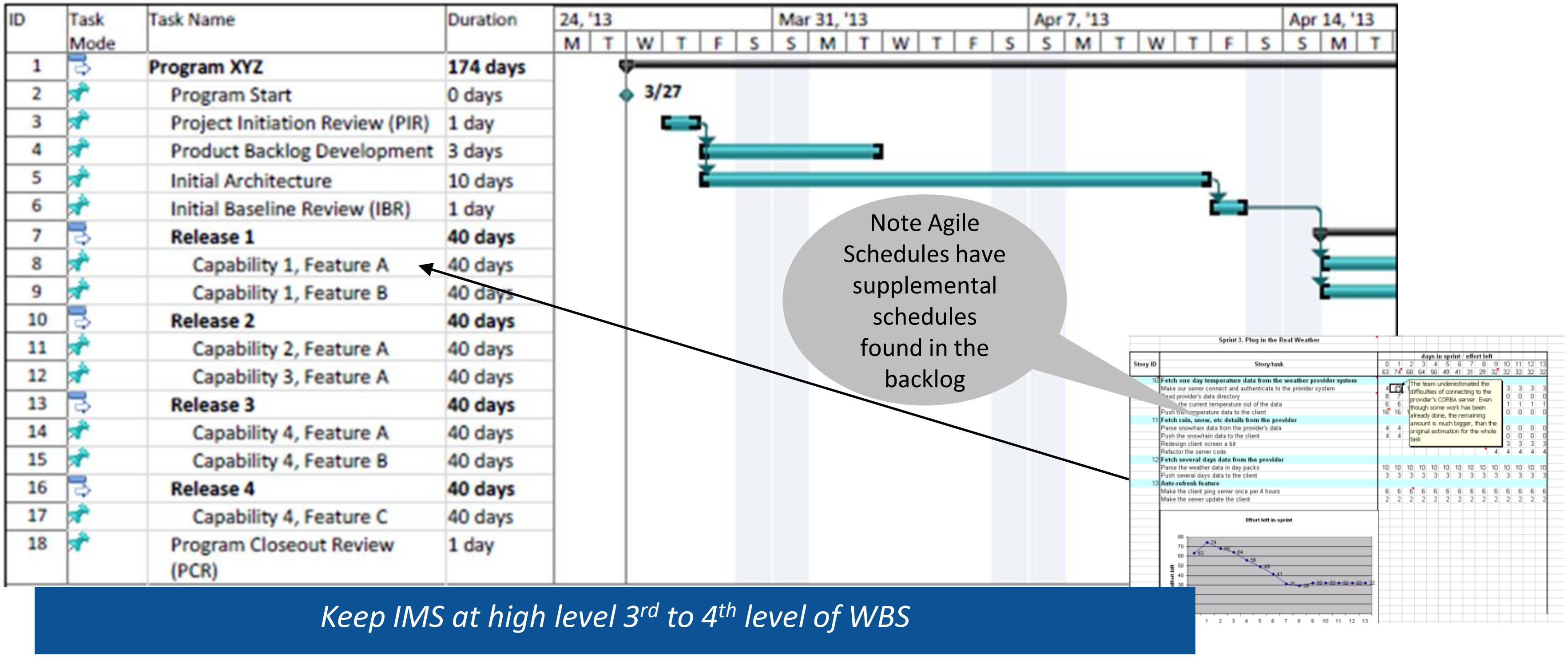


NextGenLM

Baseline current schedule

After we have created a high level schedule, we will baseline the schedule. With Agile programs schedules will continuously be revisited and monitored to ensure they are still accurate





ID	Task Mode	Task Name	Duration
1		**Program XYZ**	**174 days**
2		Program Start	0 days
3		Project Initiation Review (PIR)	1 day
4		Product Backlog Development	3 days
5		Initial Architecture	10 days
6		Initial Baseline Review (IBR)	1 day
7		**Release 1**	**40 days**
8		Capability 1, Feature A	40 days
9		Capability 1, Feature B	40 days
10		**Release 2**	**40 days**
11		Capability 2, Feature A	40 days
12		Capability 3, Feature A	40 days
13		**Release 3**	**40 days**
14		Capability 4, Feature A	40 days
15		Capability 4, Feature B	40 days
16		**Release 4**	**40 days**
17		Capability 4, Feature C	40 days
18		Program Closeout Review (PCR)	1 day

Note Agile Schedules have supplemental schedules found in the backlog

Sprint 3. Plug in the Real Weather

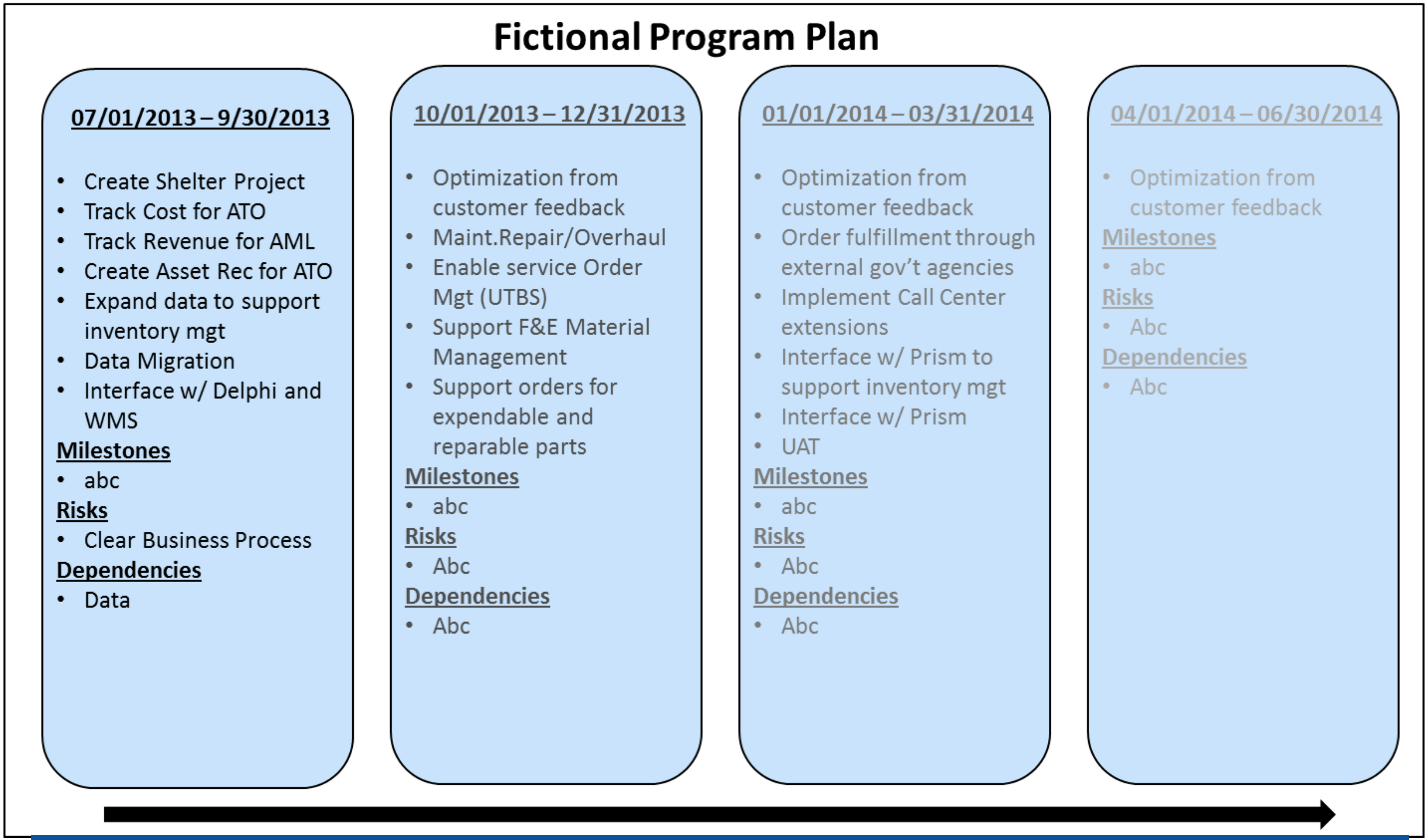
Story ID	Story/task	days in sprint / effort left													
		0	1	2	3	4	5	6	7	8	9	10	11	12	13
		63	74	68	64	56	49	41	31	29	32	32	32	32	32
10	**Fetch one day temperature data from the weather provider system**														
	Make our server connect and authenticate to the provider system	4	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	3	3	3	3
	Read provider's data directory	8	7	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0
	[illegible] the current temperature out of the data	6	6	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1	1	1	1
	Push the [illegible]mperature data to the client	16	16	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0
11	**Fetch rain, snow, etc details from the provider**														
	Parse snow/rain data from the provider's data	4	4	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0
	Push the snow/rain data to the client	4	4	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0
	Redesign client screen a bit			[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	3	3	3	3
	Refactor the server code										4	4	4	4	4
12	**Fetch several days data from the provider**														
	Parse the weather data in day packs	10	10	10	10	10	10	10	10	10	10	10	10	10	10
	Push several days data to the client	3	3	3	3	3	3	3	3	3	3	3	3	3	3
13	**Auto-refresh feature**														
	Make the client ping server once per 4 hours	6	6	6	6	6	6	6	6	6	6	6	6	6	6
	Make the server update the client	2	2	2	2	2	2	2	2	2	2	2	2	2	2

The team underestimated the difficulties of connecting to the provider's CORBA server. Even though some work has been already done, the remaining amount is much bigger, than the original estimation for the whole task

Keep IMS at high level 3rd to 4th level of WBS

NextGenLM

Agile programs plan



Fictional Program Plan

07/01/2013 – 9/30/2013

- Create Shelter Project
- Track Cost for ATO
- Track Revenue for AML
- Create Asset Rec for ATO
- Expand data to support inventory mgt
- Data Migration
- Interface w/ Delphi and WMS

Milestones

- abc

Risks

- Clear Business Process

Dependencies

- Data

10/01/2013 – 12/31/2013

- Optimization from customer feedback
- Maint.Repair/Overhaul
- Enable service Order Mgt (UTBS)
- Support F&E Material Management
- Support orders for expendable and reparable parts

Milestones

- abc

Risks

- Abc

Dependencies

- Abc

01/01/2014 – 03/31/2014

- Optimization from customer feedback
- Order fulfillment through external gov't agencies
- Implement Call Center extensions
- Interface w/ Prism to support inventory mgt
- Interface w/ Prism
- UAT

Milestones

- abc

Risks

- Abc

Dependencies

- Abc

04/01/2014 – 06/30/2014

- Optimization from customer feedback

Milestones

- abc

Risks

- Abc

Dependencies

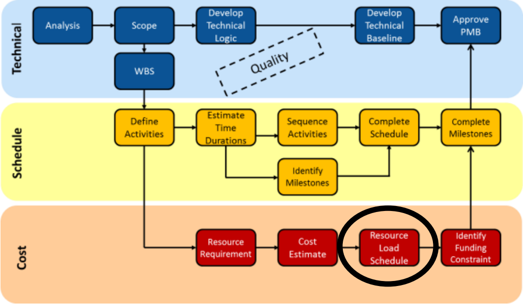
- Abc

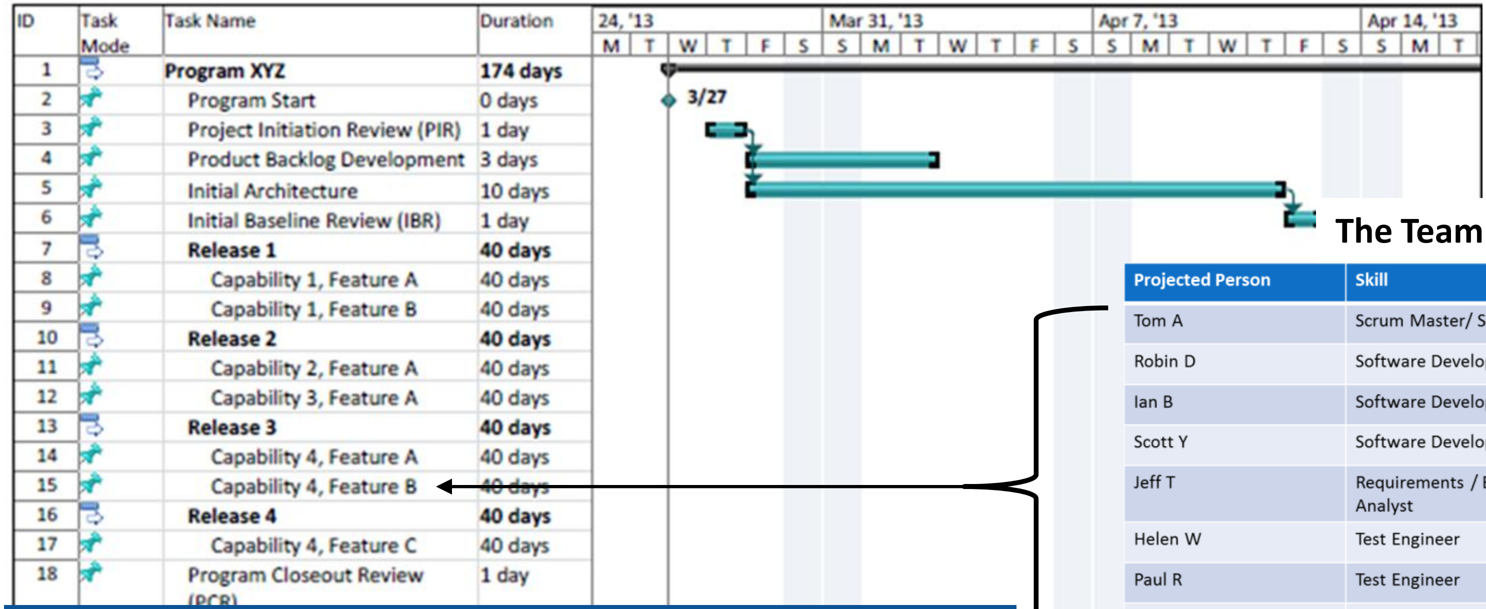
We need to have a program plan at high level

NextGenLM

Resource load

All programs need to understand their resource allocation in order to understand whether they can successfully complete the project. Agile programs load *teams* against the schedule as opposed to *individuals*. The team is responsible for completing all work needed to complete the project.





ID	Task Mode	Task Name	Duration
1		**Program XYZ**	**174 days**
2		Program Start	0 days
3		Project Initiation Review (PIR)	1 day
4		Product Backlog Development	3 days
5		Initial Architecture	10 days
6		Initial Baseline Review (IBR)	1 day
7		**Release 1**	**40 days**
8		Capability 1, Feature A	40 days
9		Capability 1, Feature B	40 days
10		**Release 2**	**40 days**
11		Capability 2, Feature A	40 days
12		Capability 3, Feature A	40 days
13		**Release 3**	**40 days**
14		Capability 4, Feature A	40 days
15		Capability 4, Feature B	40 days
16		**Release 4**	**40 days**
17		Capability 4, Feature C	40 days
18		Program Closeout Review (PCR)	1 day

The Team

Projected Person	Skill	Level
Tom A	Scrum Master/ Software	5
Robin D	Software Developer	4
Ian B	Software Developer	3
Scott Y	Software Developer	3
Jeff T	Requirements / Business Analyst	2
Helen W	Test Engineer	4
Paul R	Test Engineer	3
James B	Database Engineer	4
		3.5

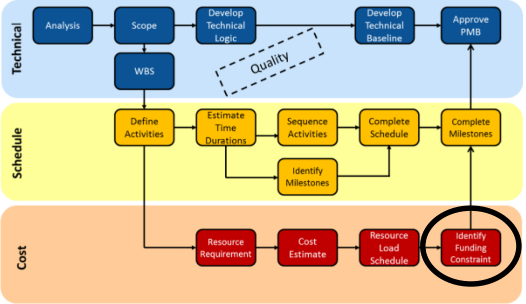
Fictional Average = $100 hour

Agile teams swarm on work, resource loading needs to be aggregated

NextGenLM

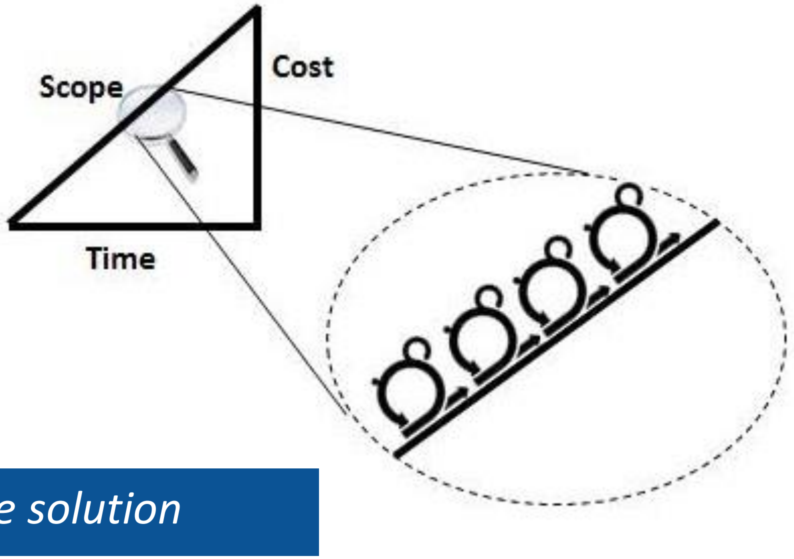
Identify any funding constraints

Before teams can complete their performance measurement baseline, they need identify and analyze any potential funding Constraints they may have.



- ✓ Contract terms and conditions
- ✓ Appropriation of funds
- ✓ Budget profiles

The benefit of Agile, is that programs can get started even if a customers funding profile does not cover the entire scope of work. Customers can purchase incremental features, with regular feedback cycles to prioritize.



Meet with contracts regarding our Agile solution

NextGenLM

Incrementally complete milestones

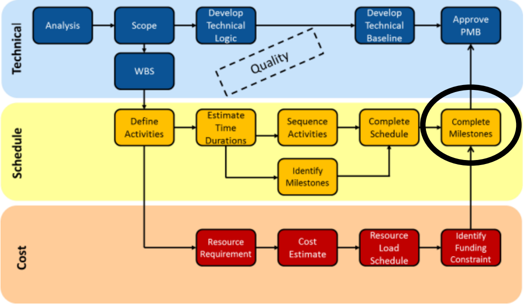
Once the baseline is instantiated, teams can begin to complete milestones incrementally, allowing us to be responsive to stakeholders changing needs.

Ensure we begin with the end in Mind, Clear acceptance criteria

What

Measurable success Criteria

- New Biometrics System
- Automates 85% of workflows
- 80% of external systems integrated
- 80% of users satisfied with interface
- Single Sign on
- 100% secure



How

ID	Feature
1	Implement Log-in
2	Automated Export to excel
3	Integrate with system x
4	Integrate with system y
….	….
180	User Preferences
181	Security Feature

Focus on the What, this aligns to business value

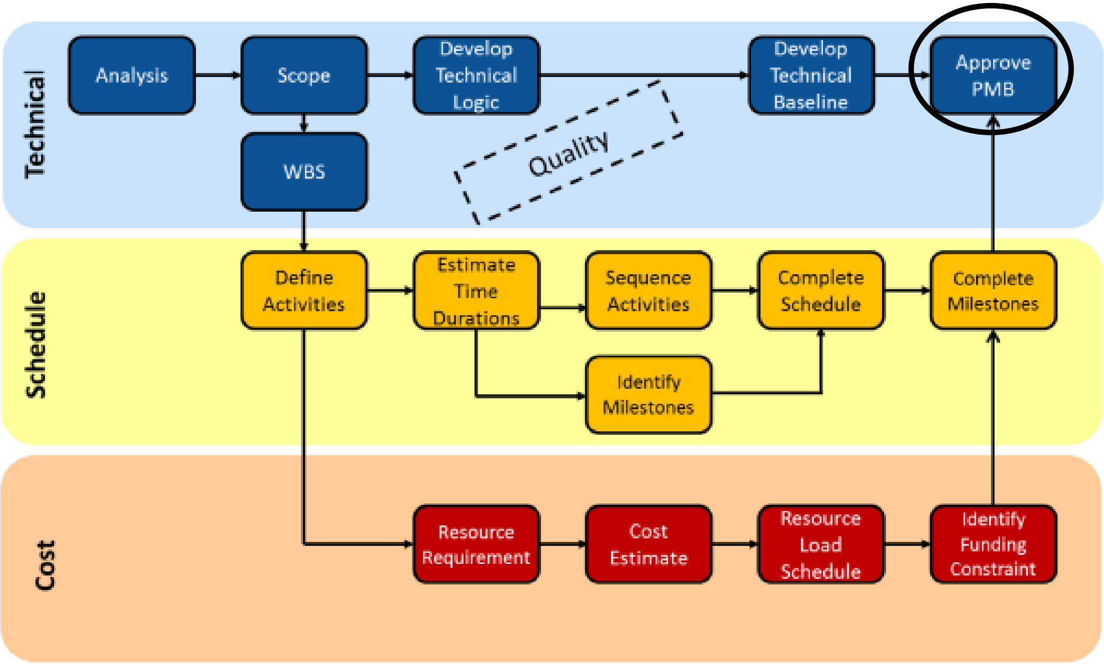
NextGenLM

Approve current baseline

All Programs have a baseline to work from and roadmap to reference.

With Agile programs we Include the team in developing maintaining, and tracking the PMB.

We start with the knowledge that life changes and include regular feedback loops to update the baseline.



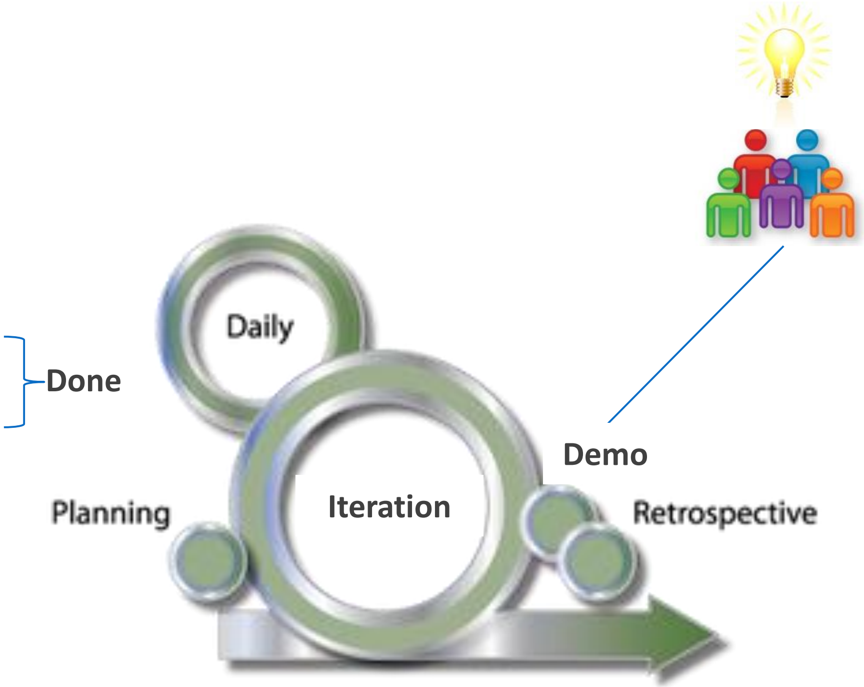
Don't forget a robust change management strategy to keep baseline current and accurate.

NextGenLM

Accommodating Change

Backlog				
Story	Title	Pri	Size	…
4	Build …			
5	Add UI ..			
6	Add …			
…				
10	Story J			
11	Story K			
12	Story L			
13	Story M			

Done



Backlog				
Story	Title	Pri	Size	…
4	Build …			
5	Add UI ..			
6	Add …			
…				
10	Story J		8	
11	Story K			
12	Story L			
13	Story M			

Out

In

New	Story N		8	

Knowledge gained from early iterations changes the content of the backlog without changing overall commitment.

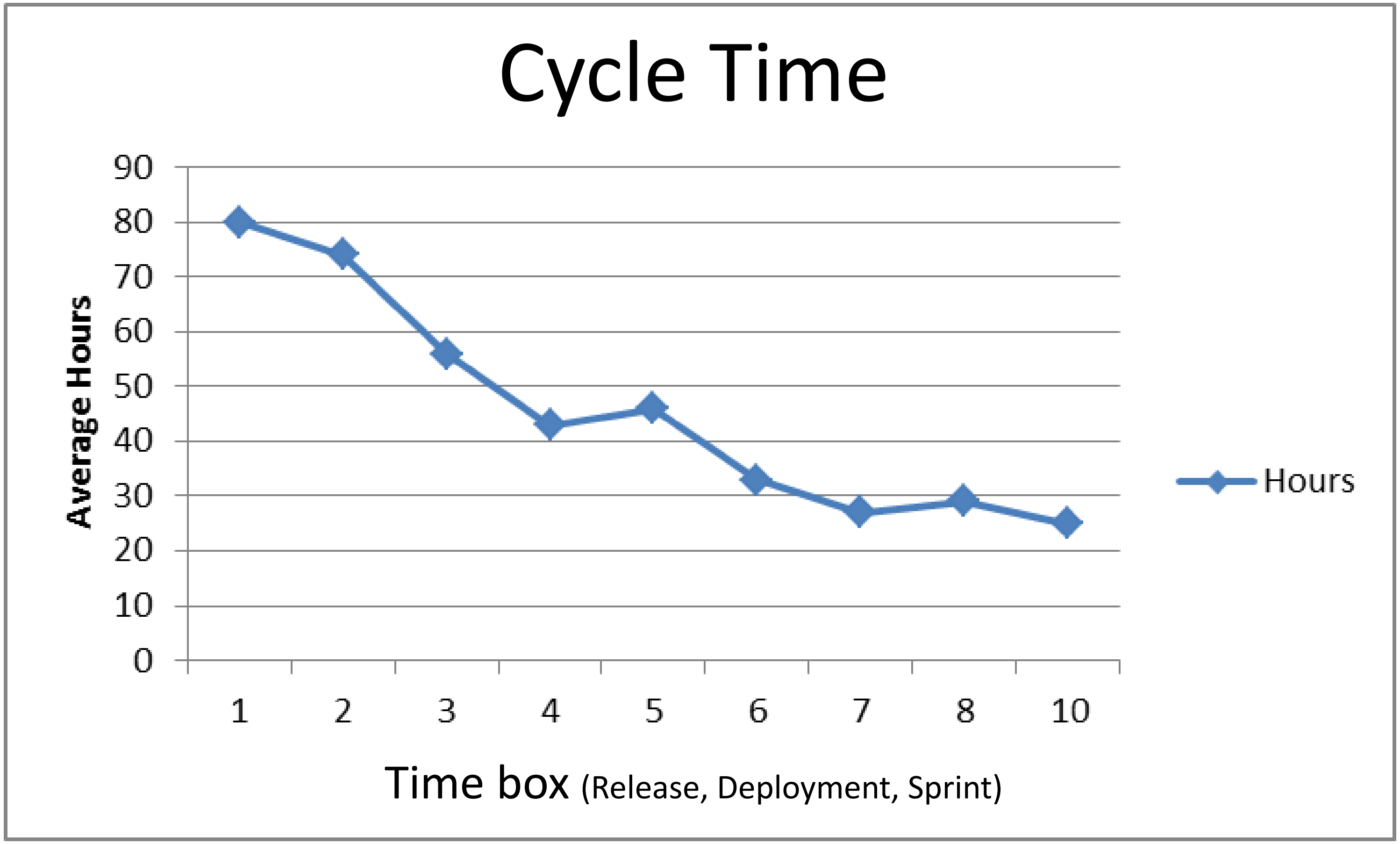
NextGenLM

Planned vs Actual

Planned	Actual
Stories	Stories Accepted
Hour	Hours
Features	Features Accepted
Release Content	Release Content
Velocity	Velocity

Planning is key in Agile, understanding how we are performing against plan is critical data.

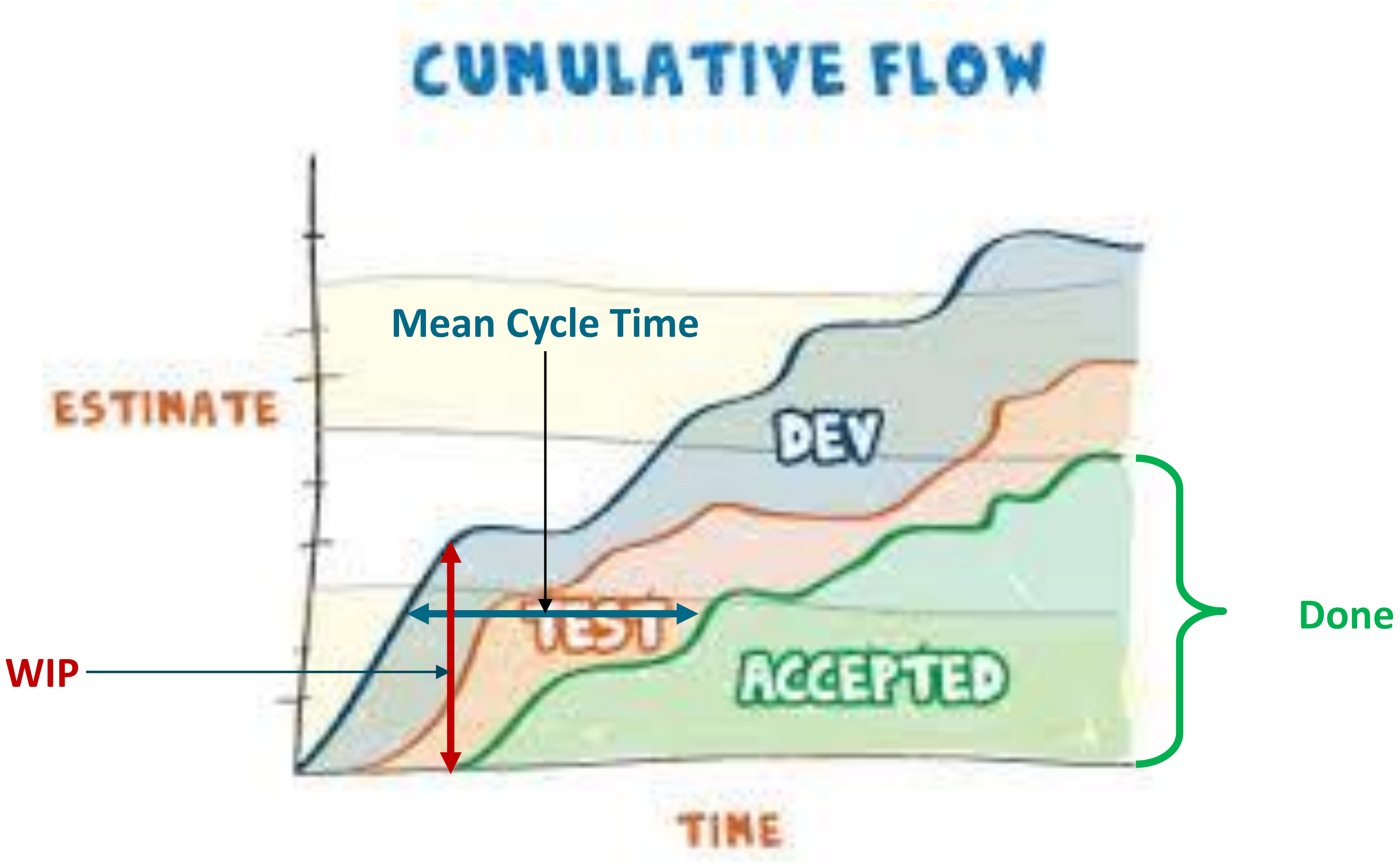
Cycle Time



Tracking cycle time allows us to visualize work and continuously improve

NextGenLM

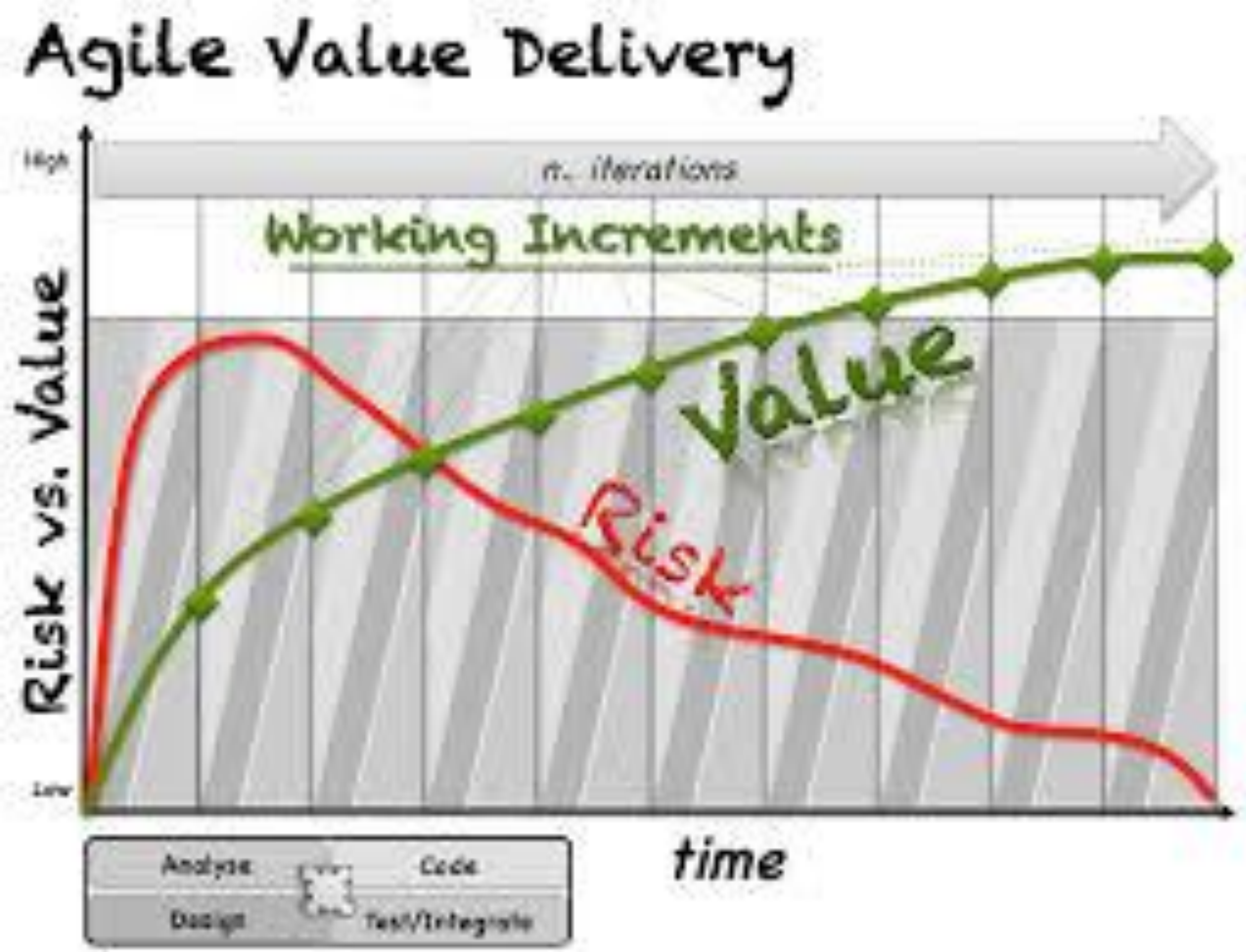
Cumulative Flow



Tracking Continuous Flow Provides a lot of data in a single diagram

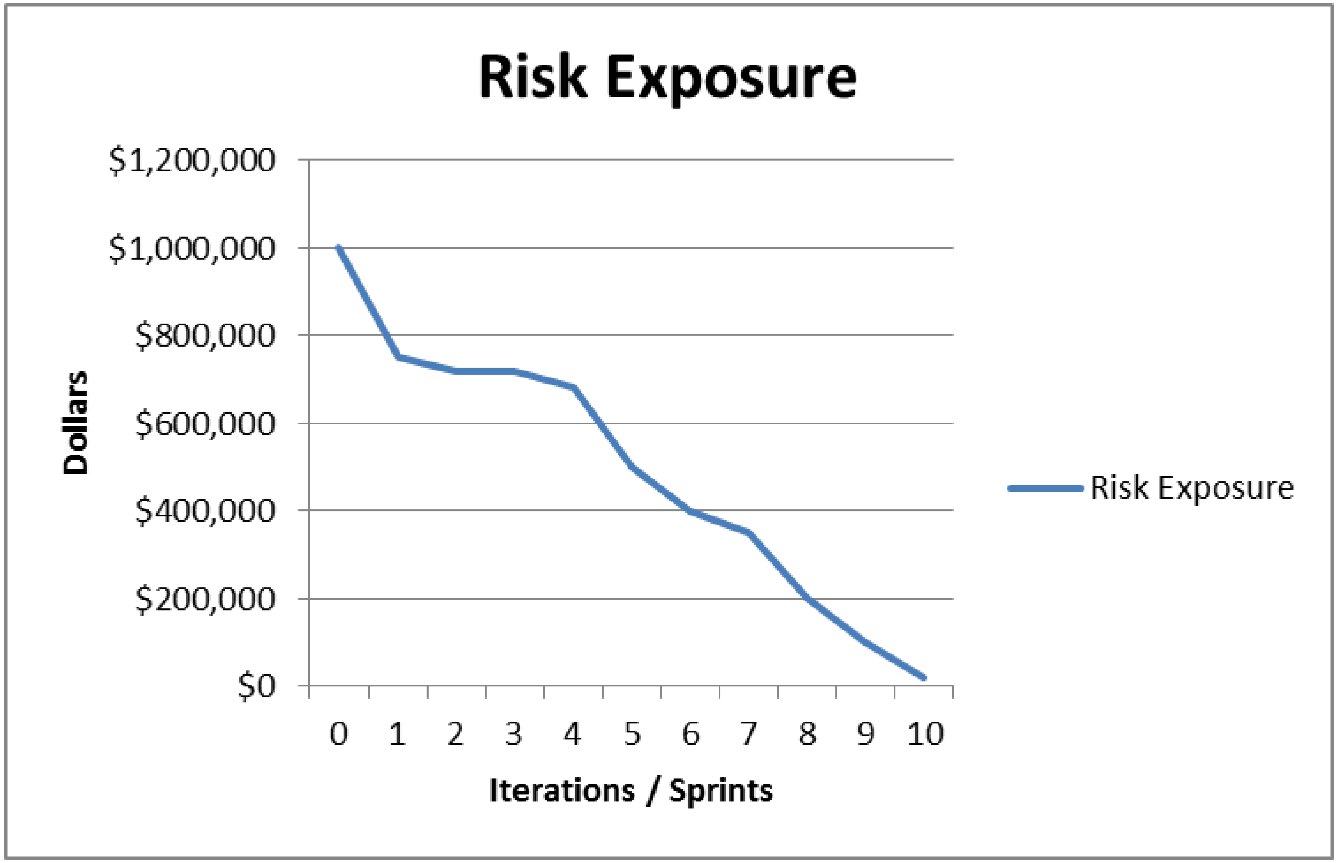
NextGenLM

Value Delivered



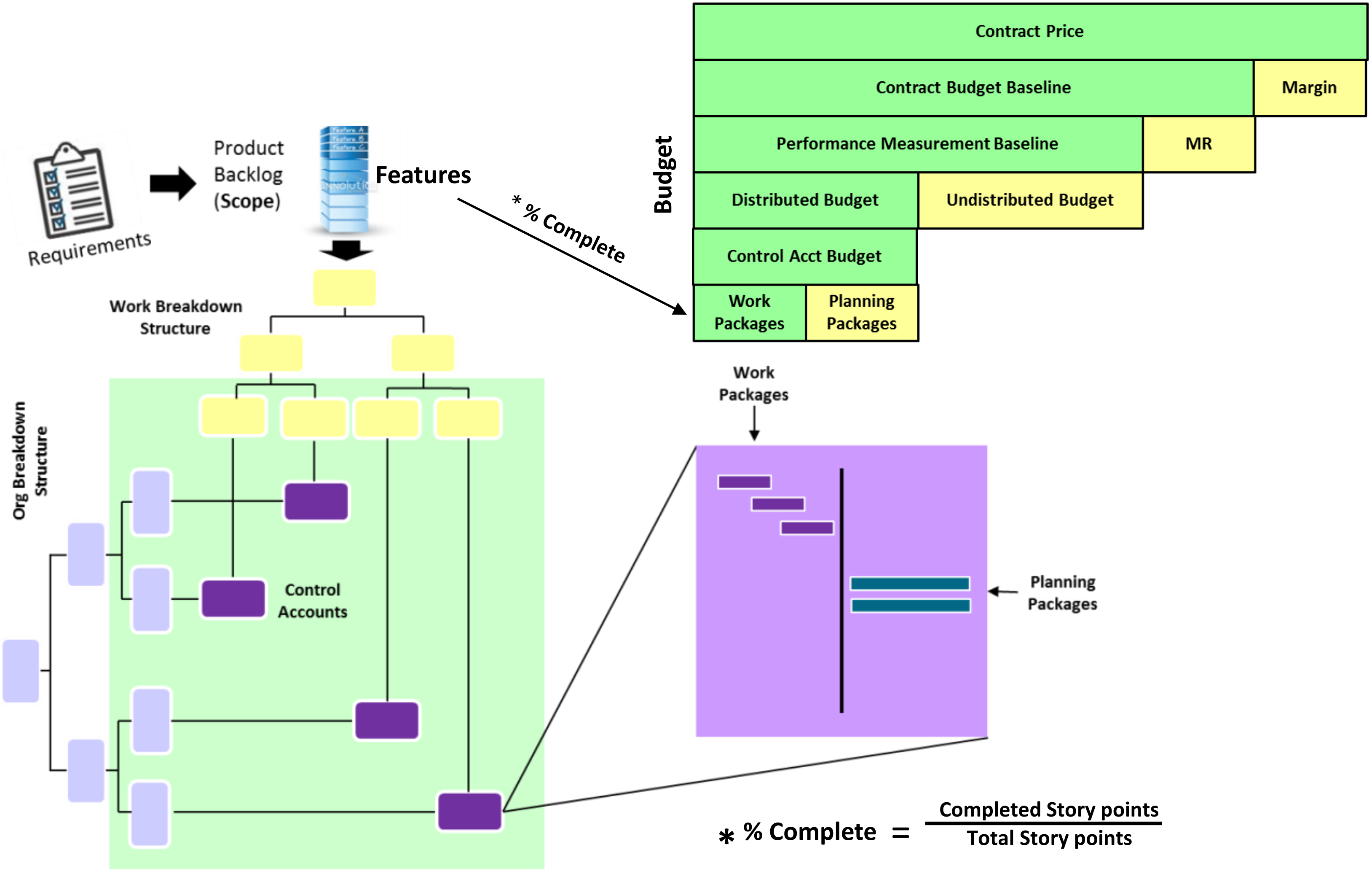
Units of Value Delivered

Risk Exposure in Dollars



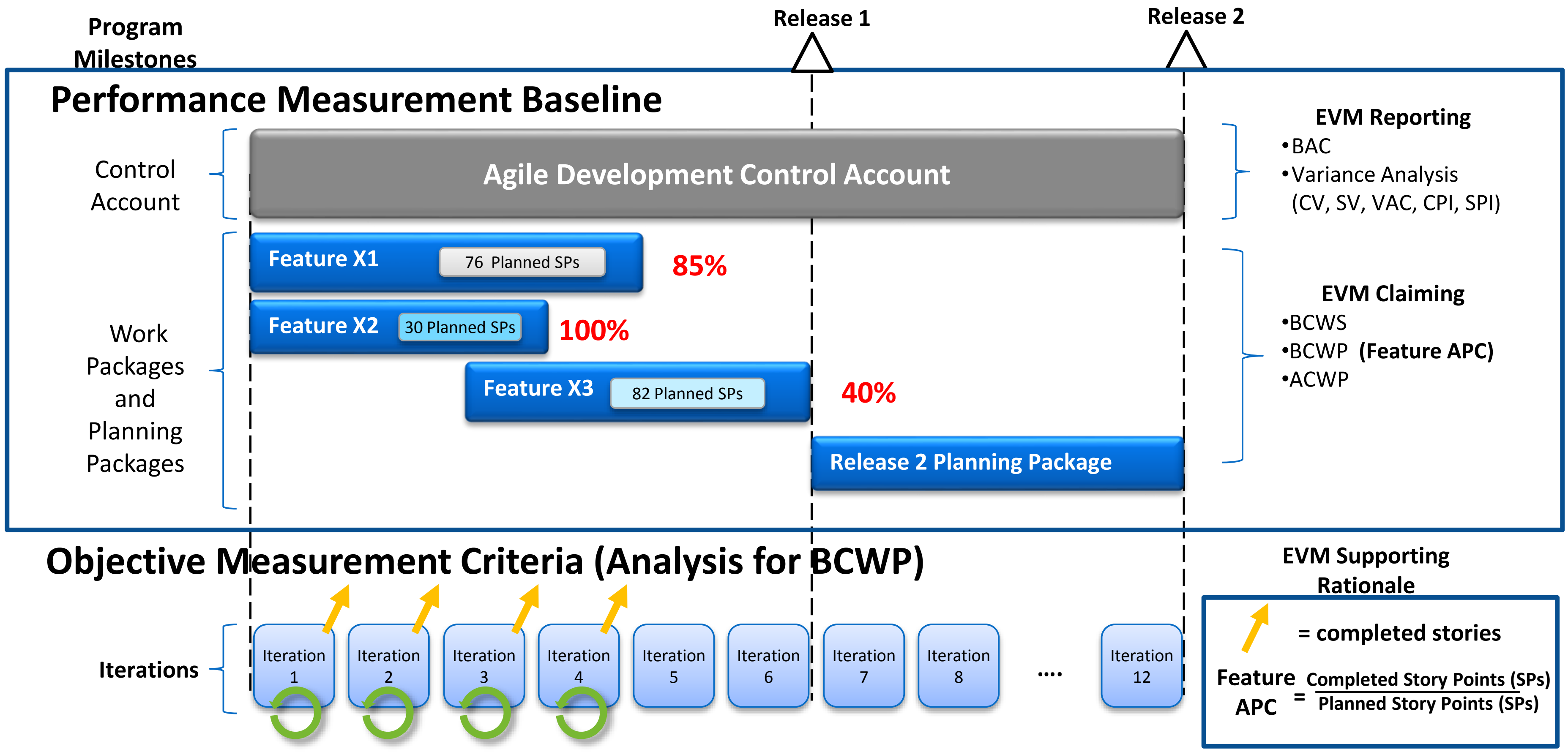
Burn down of risks and calculated exposure

EV Structure



$$* \ \% \text{ Complete} = \frac{\text{Completed Story points}}{\text{Total Story points}}$$

Claiming performance



Program Milestones

Release 1

Release 2

Performance Measurement Baseline

Control Account

Agile Development Control Account

EVM Reporting
- BAC
- Variance Analysis (CV, SV, VAC, CPI, SPI)

Work Packages and Planning Packages

Feature X1 76 Planned SPs 85%

Feature X2 30 Planned SPs 100%

Feature X3 82 Planned SPs 40%

Release 2 Planning Package

EVM Claiming
- BCWS
- BCWP **(Feature APC)**
- ACWP

Objective Measurement Criteria (Analysis for BCWP)

Iterations

Iteration 1, Iteration 2, Iteration 3, Iteration 4, Iteration 5, Iteration 6, Iteration 7, Iteration 8 …. Iteration 12

EVM Supporting Rationale

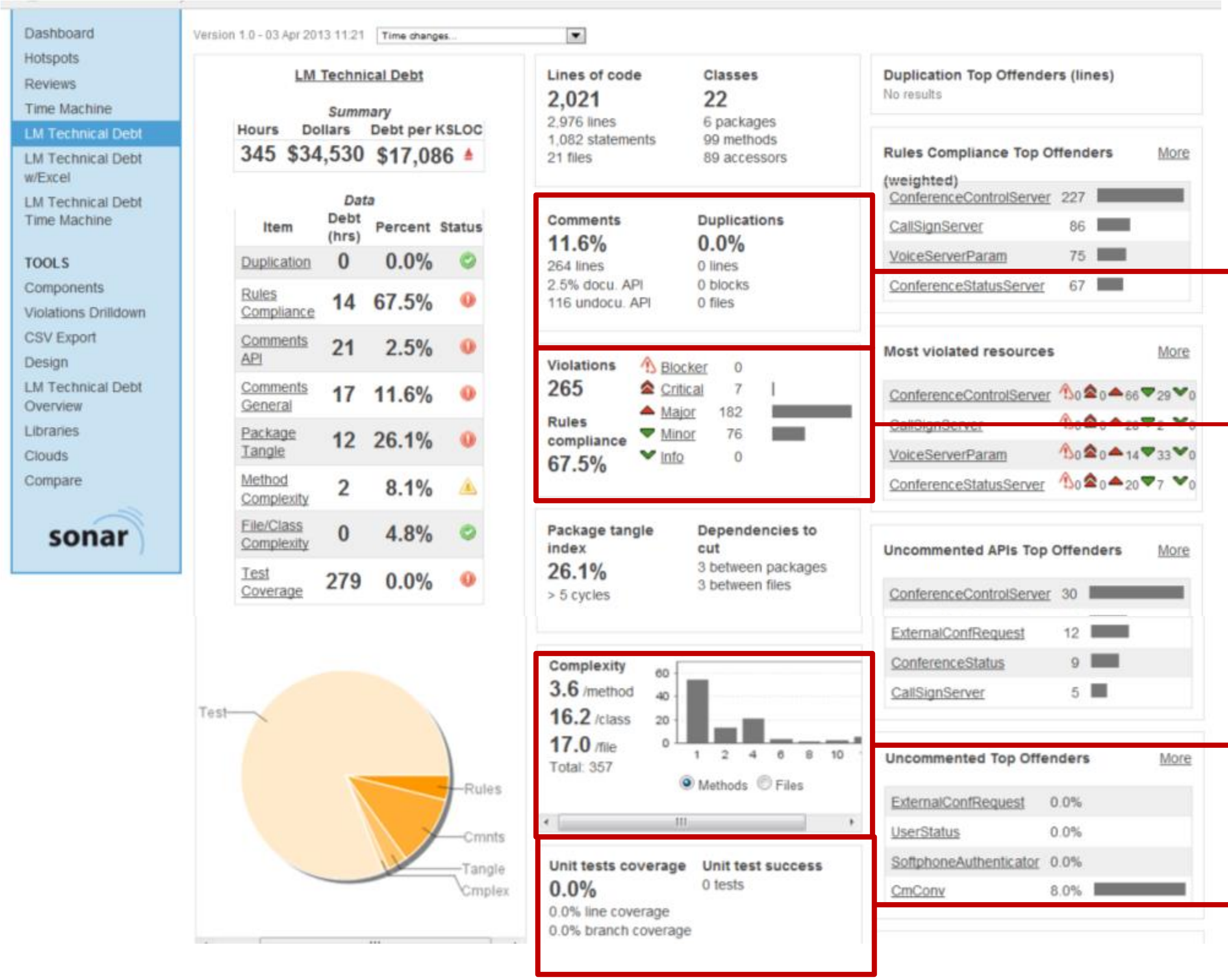
= completed stories

$$\text{Feature APC} = \frac{\text{Completed Story Points (SPs)}}{\text{Planned Story Points (SPs)}}$$

Take EVM at the Feature Level

NextGenLM

Technical Debt Metrics



Version 1.0 - 03 Apr 2013 11:21 Time changes...

LM Technical Debt

Summary

Hours	Dollars	Debt per KSLOC
345	$34,530	$17,086 ▲

Data

Item	Debt (hrs)	Percent	Status
Duplication	0	0.0%	✓
Rules Compliance	14	67.5%	!
Comments API	21	2.5%	!
Comments General	17	11.6%	!
Package Tangle	12	26.1%	!
Method Complexity	2	8.1%	⚠
File/Class Complexity	0	4.8%	✓
Test Coverage	279	0.0%	!

Lines of code
2,021
2,976 lines
1,082 statements
21 files

Classes
22
6 packages
99 methods
89 accessors

Comments
11.6%
264 lines
2.5% docu. API
116 undocu. API

Duplications
0.0%
0 lines
0 blocks
0 files

Violations
265

Rules compliance
67.5%

Blocker	0	
Critical	7	▌
Major	182	████████
Minor	76	███
Info	0	

Package tangle index
26.1%
> 5 cycles

Dependencies to cut
3 between packages
3 between files

Complexity
3.6 /method
16.2 /class
17.0 /file
Total: 357

● Methods ○ Files

Unit tests coverage
0.0%
0.0% line coverage
0.0% branch coverage

Unit test success
0 tests

Duplication Top Offenders (lines)
No results

Rules Compliance Top Offenders (weighted) More

ConferenceControlServer	227
CallSignServer	86
VoiceServerParam	75
ConferenceStatusServer	67

Most violated resources More

	Blocker	Critical	Major	Minor	Info
ConferenceControlServer	0	0	66	29	0
CallSignServer	0	0	28	2	0
VoiceServerParam	0	0	14	33	0
ConferenceStatusServer	0	0	20	7	0

Uncommented APIs Top Offenders More

ConferenceControlServer	30
ExternalConfRequest	12
ConferenceStatus	9
CallSignServer	5

Uncommented Top Offenders More

ExternalConfRequest	0.0%
UserStatus	0.0%
SoftphoneAuthenticator	0.0%
CmConv	8.0%

Percentage of duplicated code

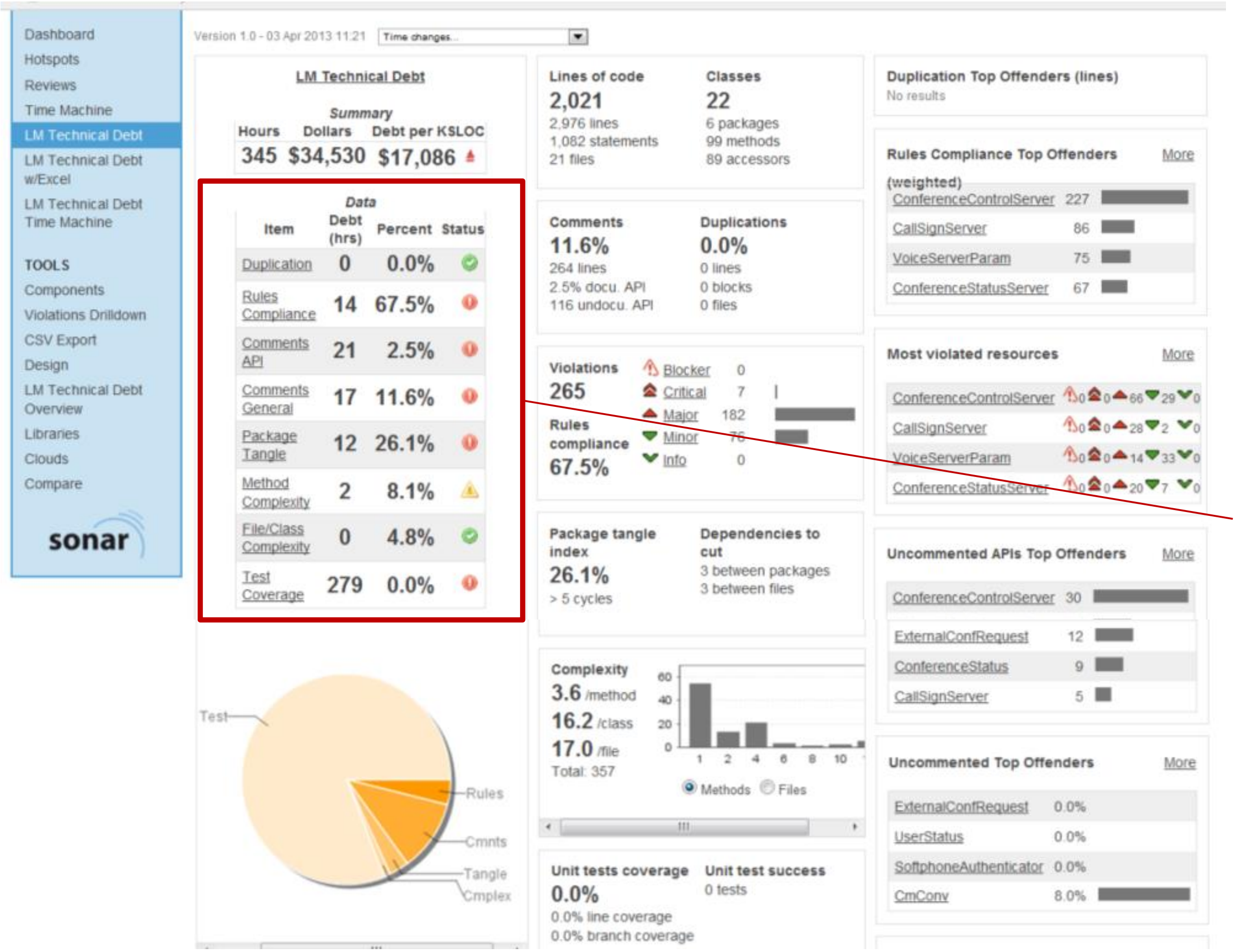
Check for error patterns

Cycylomatic Complexity

Test Coverage

NextGenLM

Technical Debt Metrics



Version 1.0 - 03 Apr 2013 11:21

LM Technical Debt

Summary

Hours	Dollars	Debt per KSLOC
345	$34,530	$17,086 ▲

Data

Item	Debt (hrs)	Percent	Status
Duplication	0	0.0%	✓
Rules Compliance	14	67.5%	!
Comments API	21	2.5%	!
Comments General	17	11.6%	!
Package Tangle	12	26.1%	!
Method Complexity	2	8.1%	⚠
File/Class Complexity	0	4.8%	✓
Test Coverage	279	0.0%	!

Lines of code
2,021
2,976 lines
1,082 statements
21 files

Classes
22
6 packages
99 methods
89 accessors

Comments
11.6%
264 lines
2.5% docu. API
116 undocu. API

Duplications
0.0%
0 lines
0 blocks
0 files

Violations
265

Rules compliance
67.5%

Blocker	0
Critical	7
Major	182
Minor	76
Info	0

Package tangle index
26.1%
> 5 cycles

Dependencies to cut
3 between packages
3 between files

Complexity
3.6 /method
16.2 /class
17.0 /file
Total: 357

Unit tests coverage
0.0%
0.0% line coverage
0.0% branch coverage

Unit test success
0 tests

Duplication Top Offenders (lines)
No results

Rules Compliance Top Offenders (weighted)

ConferenceControlServer	227
CallSignServer	86
VoiceServerParam	75
ConferenceStatusServer	67

Most violated resources

	Blocker	Critical	Major	Minor	Info
ConferenceControlServer	0	0	66	29	0
CallSignServer	0	0	28	2	0
VoiceServerParam	0	0	14	33	0
ConferenceStatusServer	0	0	20	7	0

Uncommented APIs Top Offenders

ConferenceControlServer	30
ExternalConfRequest	12
ConferenceStatus	9
CallSignServer	5

Uncommented Top Offenders

ExternalConfRequest	0.0%
UserStatus	0.0%
SoftphoneAuthenticator	0.0%
CmConv	8.0%

Provides a comprehensive view of internal code quality and maintainability.

Monitoring of trends for individual measures and overall debt can be more important than the absolute values.

NextGenLM

Questions

Contact Information:
robin.yeman@lmco.com
571-535-5854