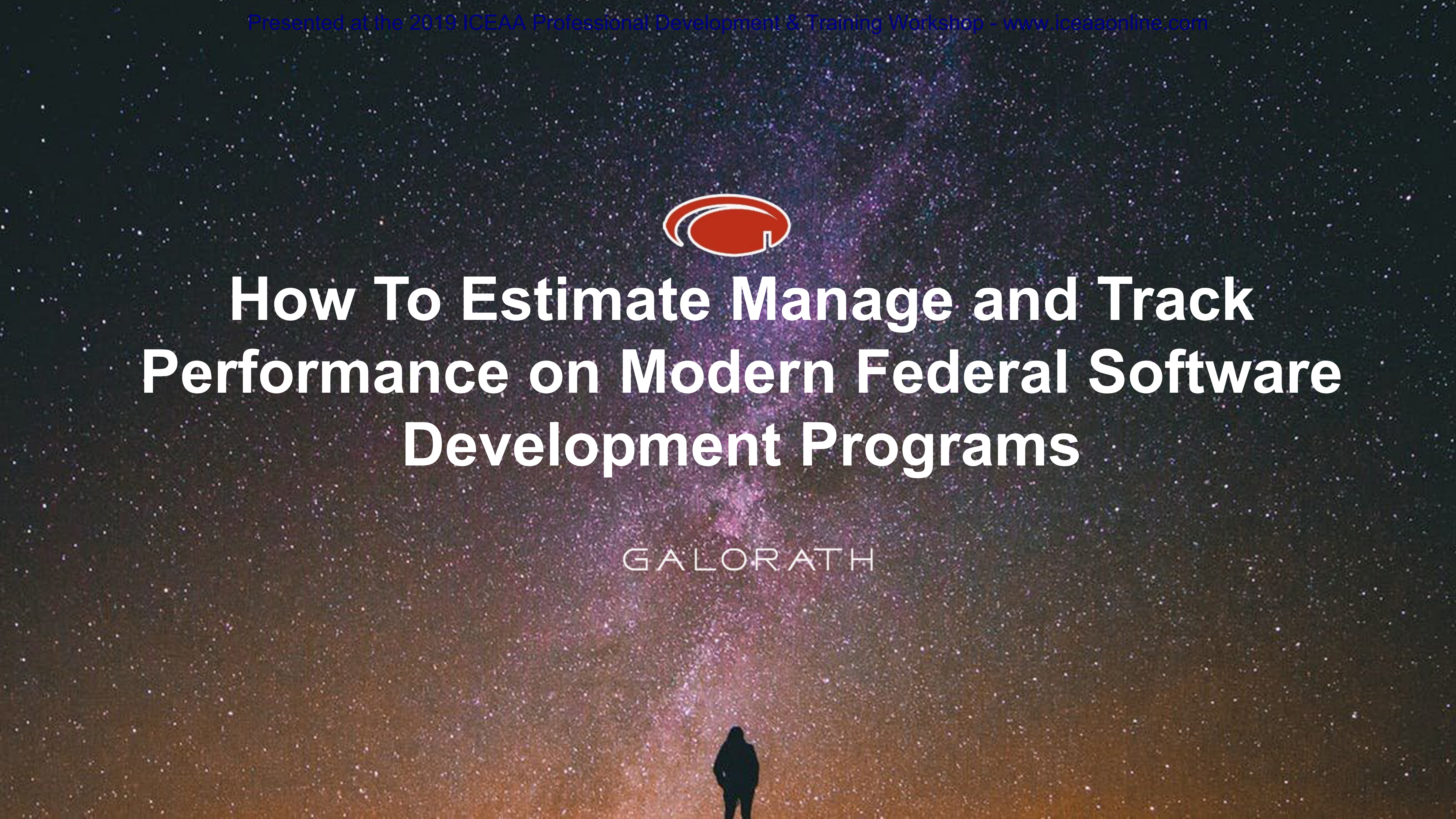


How To Estimate Manage and Track Performance on Modern Federal Software Development Programs

GALORATH

AGENDA FOR TODAY

- Cost and Schedule Overruns
- Defense Science Board Recommendations
- #NoEstimates
- Agile Estimation
- Significant Reasons for Software Cost Growth
- Managing Modern Software Development Programs
- Key Questions to Assess the Quality of the Agile Progress
- Earned Value
- Conclusions.

COST AND SCHEDULE OVERUNS

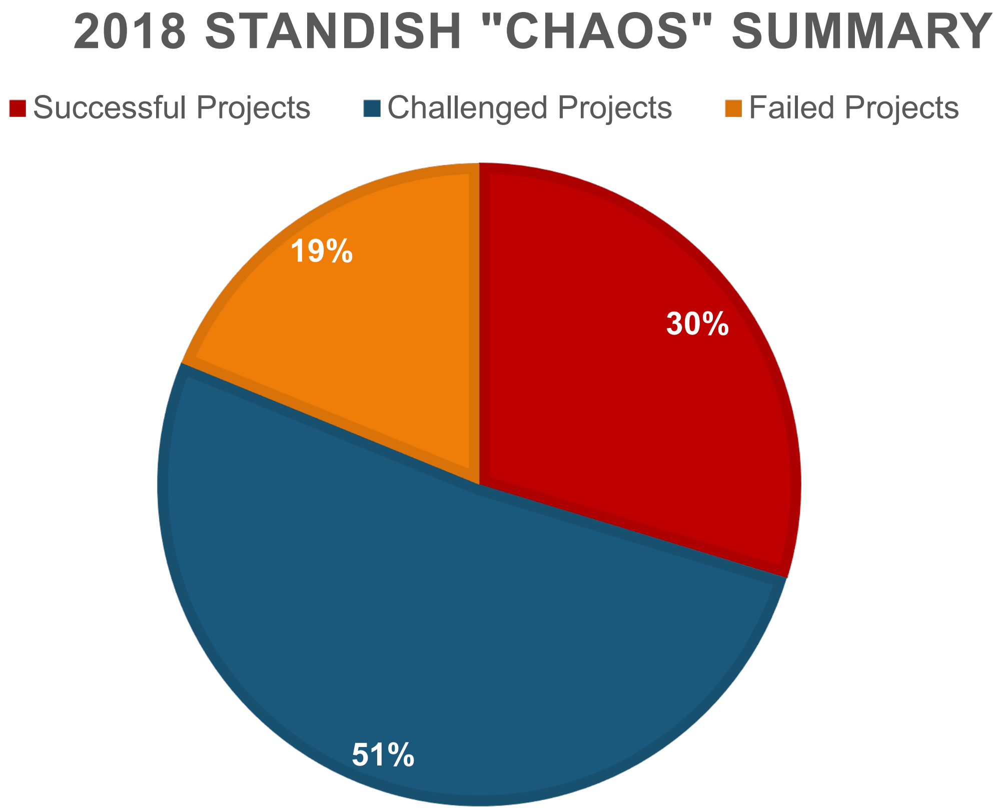
Many studies attempting to quantify the cost of software failures.

Generally agree that the number is around $50 to $80 billion annually.

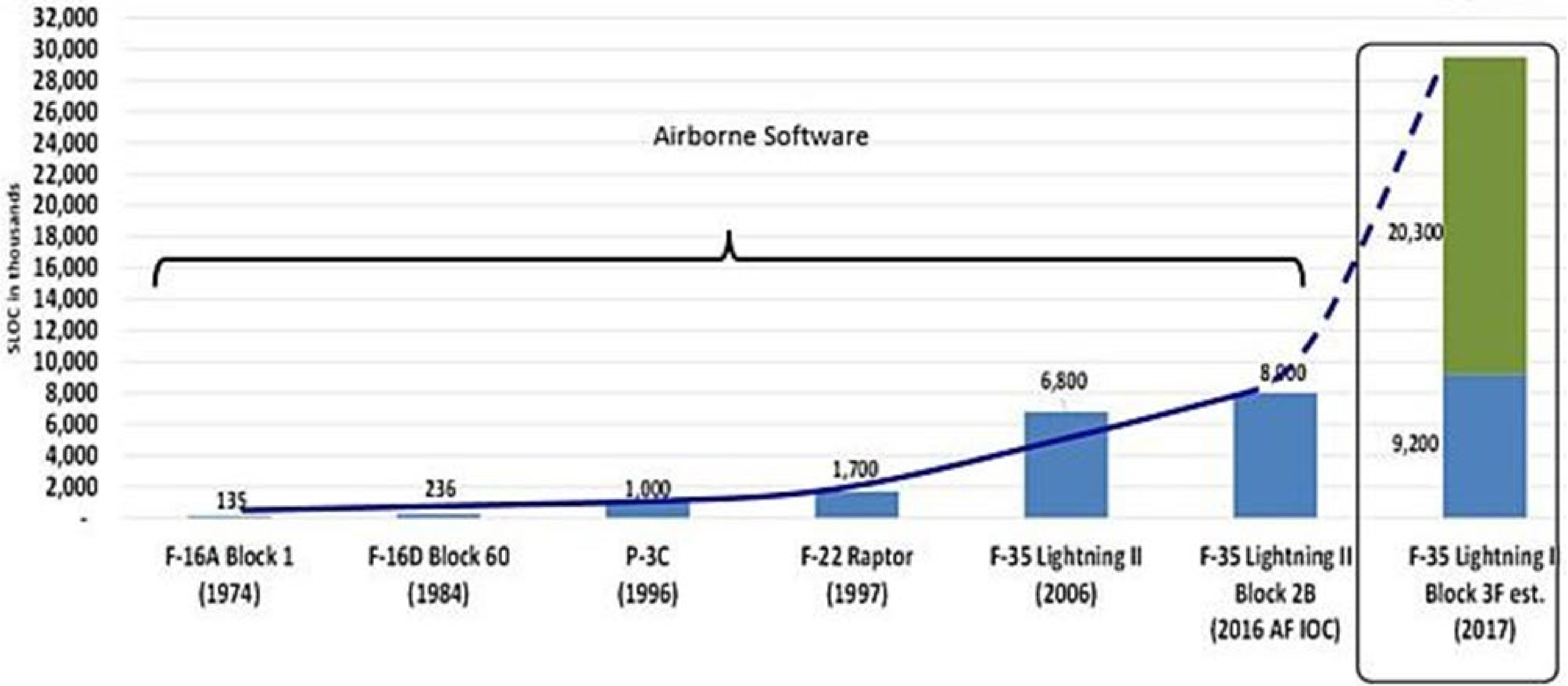
The Standish Chaos Report, which is probably the most well-known of these studies, defines success as projects delivered within budget, on schedule, and with expected functionality.

The 2018 Chaos report shows:

- Successful Projects: 30%
- Challenged Projects: 52%
- Failed Projects: 19%



SOFTWARE GROWTH IN AIRCRAFT SYSTEMS



DEFENSE SCIENCE BOARD* Recommendation

Recommendation 1: Implement the “Software Factory”

Recommendation 2: Adopt continuous iterative development best practices (continuing through sustainment) for software.

Recommendation 3: Implement: Multiple vendors to begin work with down select; Service cost estimators should modernize cost/schedule estimation processes; Project manager should build program-appropriate frameworks for status metrics; examples include: sprint burndown, epic and release burndown, velocity trending, control chart, line of balance and cumulative flow diagrams.

Recommendation 4: Current and legacy programs should plan transition to a software factory and continuous iterative development

Recommendation 5: Develop a modern software development expertise

Recommendation 6: Software is Immortal

Recommendation 7: Implement Independent Validation and Verification (IV&V)

* Final Report of the U.S. Defense Science Board (DSB) Task Force on the Design and Acquisition of Software for Defense Systems, February 2018

AGILE ESTIMATES

Are they necessary?

#NOESTIMATES

- Estimates are difficult to produce
- Provide little to no value
- Estimation is overhead and should be minimized

.

#ESTIMATES

- Organizations need to do budget planning
- Estimates are needed to make informed decisions
- Managers need estimates for accountability to shareholders

UNDERLYING TRUTHS

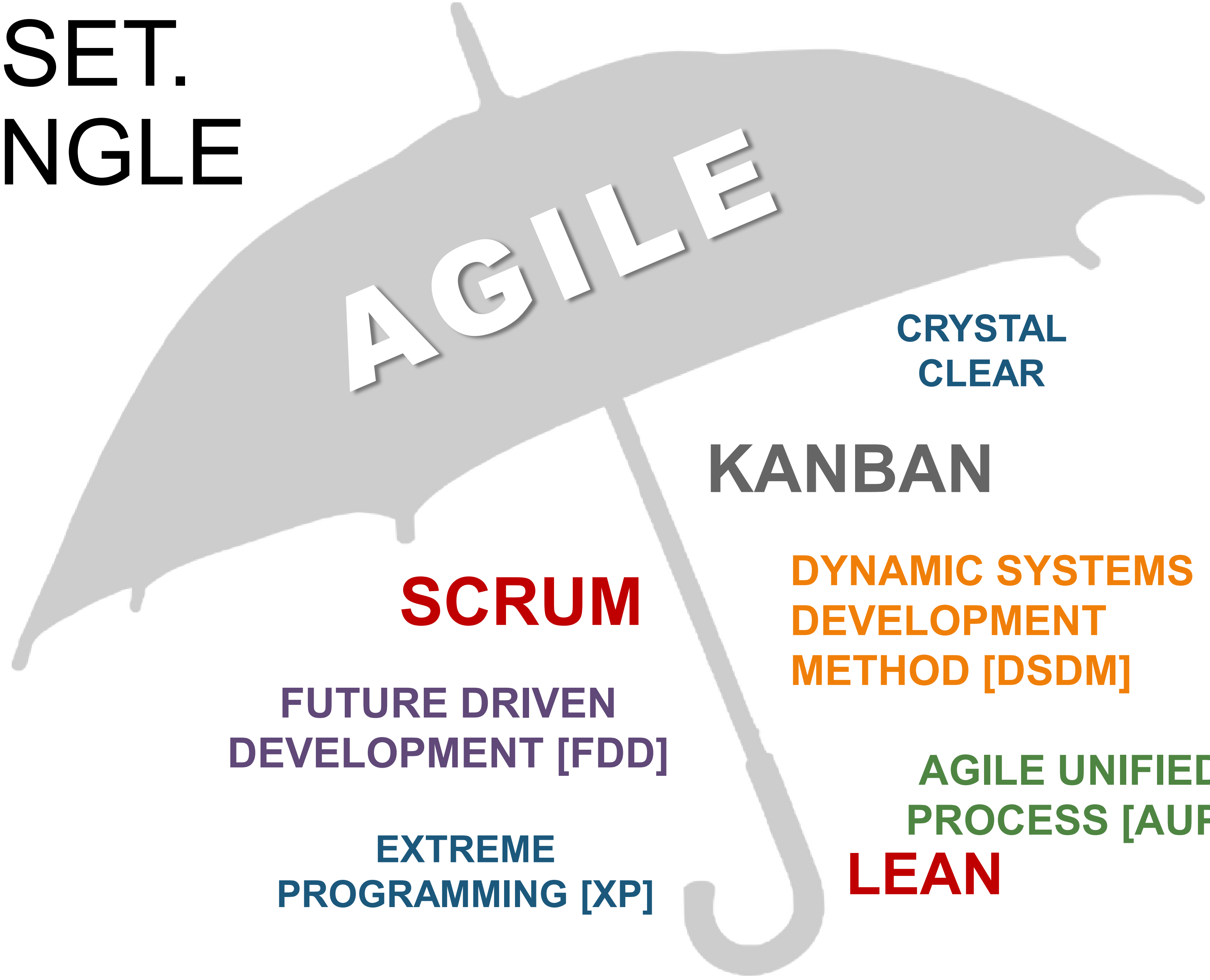
Software estimation is challenging, Agile developers see estimates as committing them to a schedule and therefore they are antithetical to the Agile Manifesto.

Software estimates drive decision making - they are not just for the developers

Total ownership cost should be considered for the immortal systems and as a result, more emphasis and research should be and is being applied to the area of software maintenance.

.

AGILE MINDSET. NOT ONE SINGLE METHOD



CRYSTAL CLEAR

KANBAN

SCRUM

DYNAMIC SYSTEMS DEVELOPMENT METHOD [DSDM]

FUTURE DRIVEN DEVELOPMENT [FDD]

AGILE UNIFIED PROCESS [AUP]

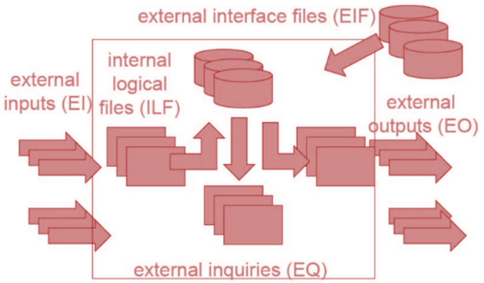
EXTREME PROGRAMMING [XP]

LEAN

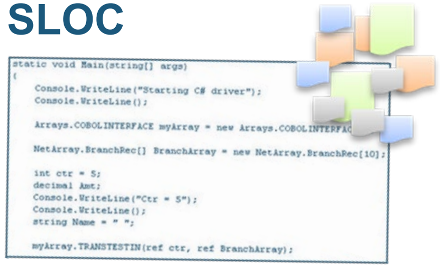
SIZE CONTINUES TO BE MAIN DRIVER

Your Subtitle Will Go Here

FUNCTION POINTS



SLOC



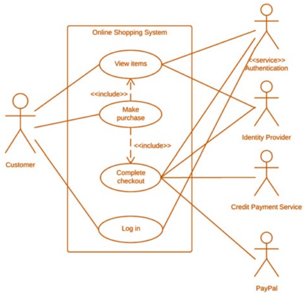
```
static void Main(string[] args)
{
    Console.WriteLine("Starting C# driver");
    Console.WriteLine();

    Arrays.COBOLINTERFACE myArray = new Arrays.COBOLINTERFAC

    NetArray.BranchRec[] BranchArray = new NetArray.BranchRec[10];

    int ctr = 5;
    decimal Amt;
    Console.WriteLine("Ctr = 5");
    Console.WriteLine();
    string Name = " ";

    myArray.TRANSTESTIN(ref ctr, ref BranchArray);
```

CHOOSE A METRIC AND BE CONSISTENT!

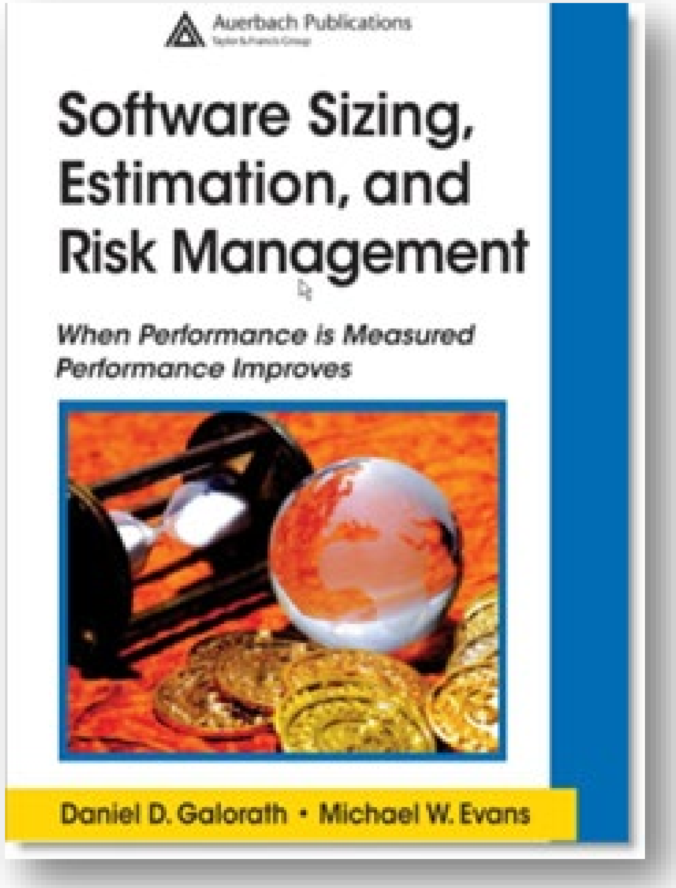
STORY POINTS

USER STORIES

USE CASES



T-SHIRT SIZING



Auerbach Publications

Software Sizing, Estimation, and Risk Management

When Performance is Measured Performance Improves

Daniel D. Galorath • Michael W. Evans

“When performance is measured performance improves”

Estimation processes are independent of tools

DRIVING THE STATE OF THE ART

10 STEP ESTIMATION PROCESS

01 Establish Estimate Scope

02 Establish Technical Baseline, Ground Rules, Assumptions

03 Collect Data

04 Estimate and Validate Software Size

05 Prepare Baseline Estimates

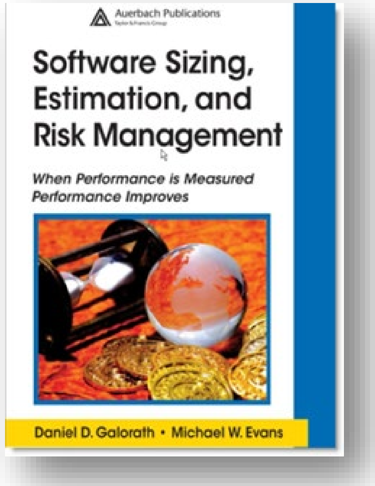
06 Review, Verify and Validate Estimate

07 Quantify Risks and Risk Analysis

08 Generate a Project Plan

09 Document Estimate and Lessons Learned

10 Track Project Throughout Development



Auerbach Publications
Taylor & Francis Group

Software Sizing, Estimation, and Risk Management

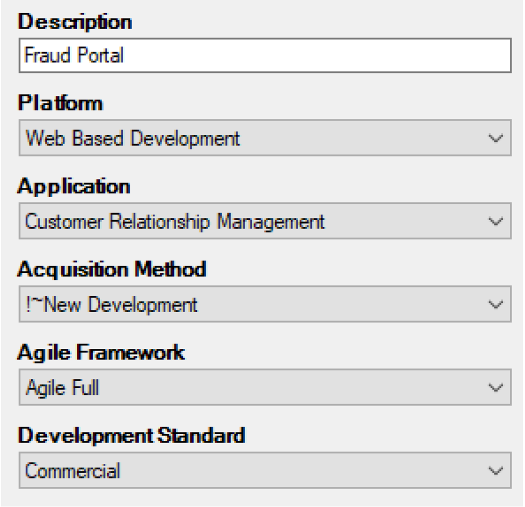
When Performance is Measured Performance Improves

Daniel D. Galorath • Michael W. Evans

Parametric Estimation for Agile Projects

Features

Project Characteristics



Description

Fraud Portal

Platform

Web Based Development

Application

Customer Relationship Management

Acquisition Method

!~New Development

Agile Framework

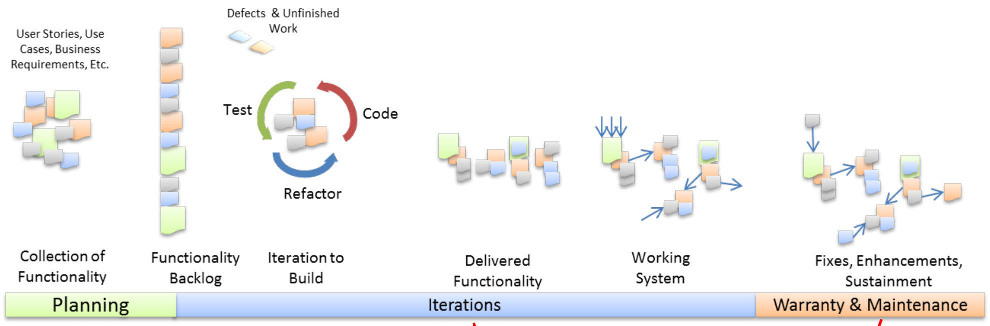
Agile Full

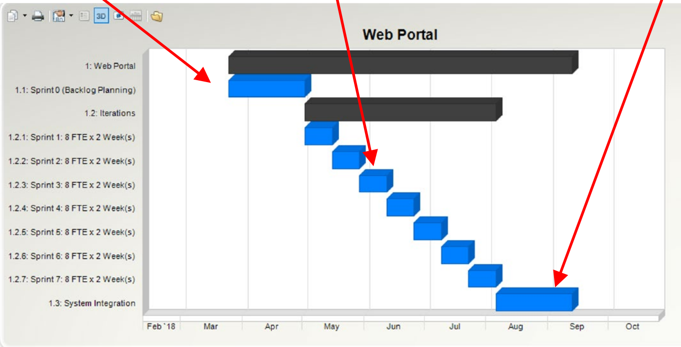
Development Standard

Commercial

Team Dynamics

Development Team


User Stories, Use Cases, Business Requirements, Etc.
Defects & Unfinished Work
Test
Code
Refactor
Collection of Functionality
Functionality Backlog
Iteration to Build
Delivered Functionality
Working System
Fixes, Enhancements, Sustainment
Planning
Iterations
Warranty & Maintenance



	Estimate
TE x 4 Weeks	
	Team A
Sprint Number	1
Sprint Duration Months	0.92
Sprint Effort Months	6.46
Sprint Effort Hours	981
Sprint Labor Cost	134,290
Sprint Velocity (UFP)	103.00
Team Size	7.00
Start Date	12/27/2018
End Date	1/24/2019

SEER AGILE PLANNER

FOR AGILE PROJECTS

DEVELOPMENT TEAM SIZE

Optimal Size can be calculated

Everyone available on day 1?

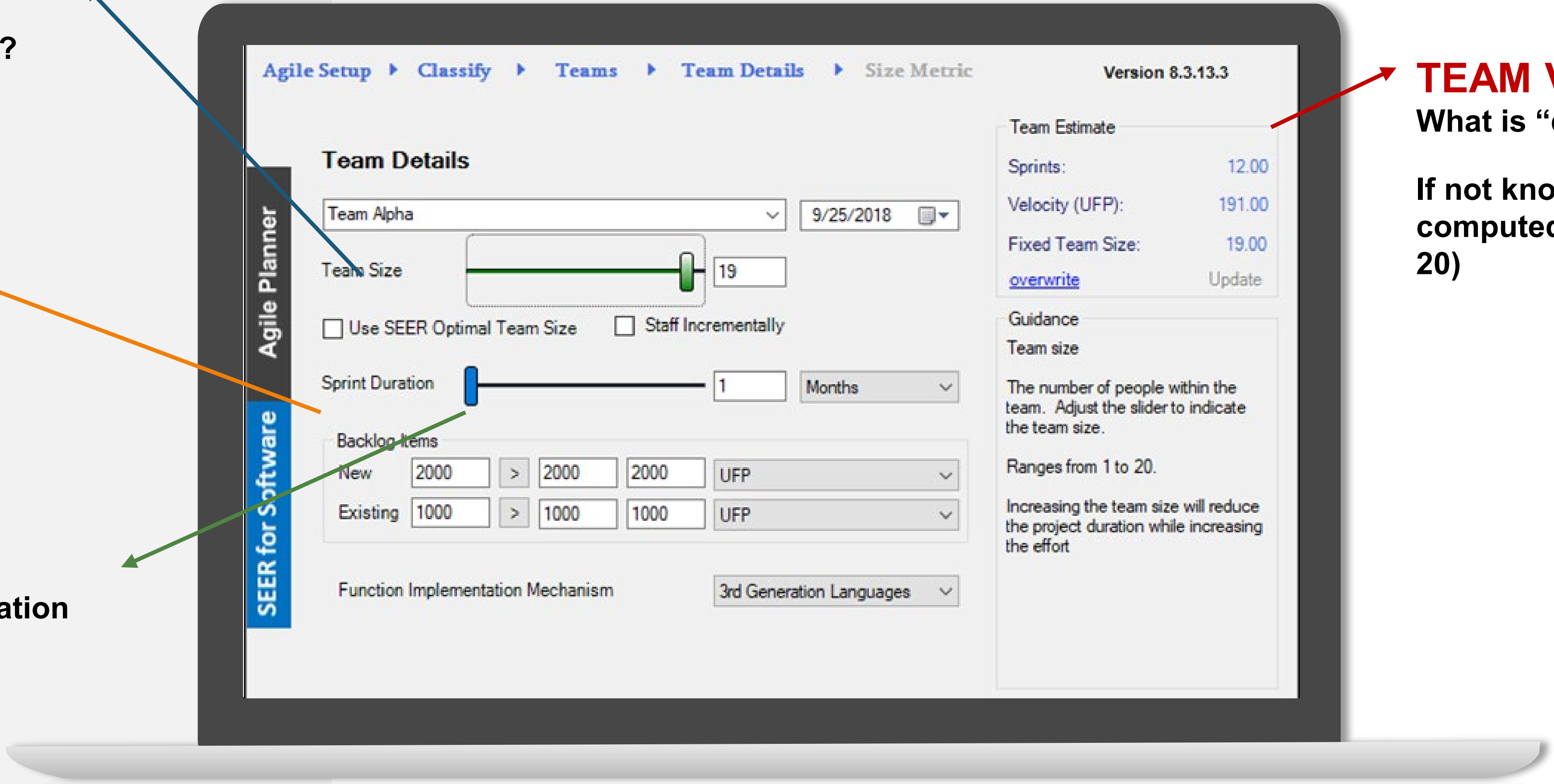
BACKLOG SIZE

What is the team tasked to complete?

SPRINT DURATION

Timebox duration for an iteration

4 and 2 weeks are the most common



Version 8.3.13.3

Team Details

Team Alpha 9/25/2018

Team Size 19

□ Use SEER Optimal Team Size □ Staff Incrementally

Sprint Duration 1 Months

Backlog Items

New	2000	>	2000	2000	UFP
Existing	1000	>	1000	1000	UFP

Function Implementation Mechanism 3rd Generation Languages

Team Estimate

Sprints:	12.00
Velocity (UFP):	191.00
Fixed Team Size:	19.00
overwrite	Update

Guidance

Team size

The number of people within the team. Adjust the slider to indicate the team size.

Ranges from 1 to 20.

Increasing the team size will reduce the project duration while increasing the effort

TEAM VELOCITY

What is "doable" in a SPRINT

If not known, it can be computed (discussed in slide 20)

ESTIMATING METHODOLOGIES

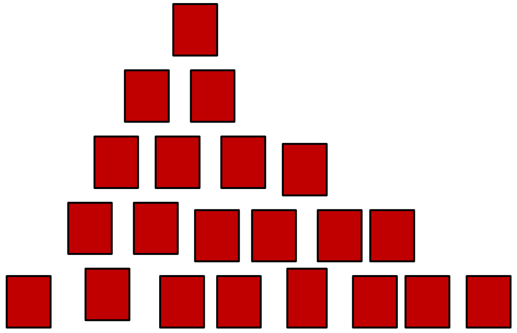
Methodology 1: Many Agile programs are fixed price, it is often just a matter of labor rates times quantity

Methodology 2: Simple Build-up approach based on averages can be defined as:Sprint Team Size (SS) x Sprint length (Sp time) x Number of Sprints (# Sprints)

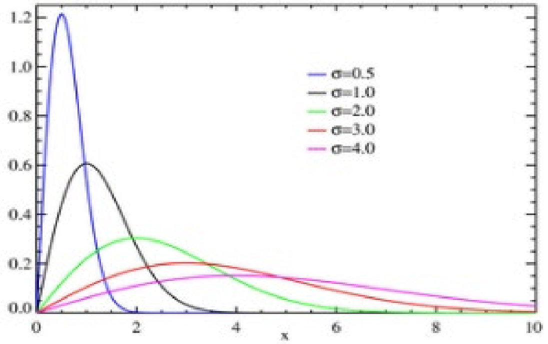
Methodology 3: Structured approach based on established “velocity” – most often used internally by the developer since detailed/sensitive data are available to them

Methodology 4: Automated Models approach based on a size metric – which may be difficult to quantify

- There is a fixed relationship between size and effort, e.g. (Effort**n)*Time = Size/Technology
- Results are then modified by current trends and analyses
- Total effort can be distributed by a mathematical model; e.g. Weibull, Rayleigh



The Sprint work projections often follow the Weibull or Rayleigh distribution



Methodology 5: Factor/Complexity approach based on data generated in early *iterations*

SOFTWARE COST GROWTH

Significant Reasons for Software Cost Growth

01 Scope Creep
Requirements Growth

02 Poor Input to Estimate

03 Failure to Clearly Define the Initial Scope

04 Unrealistic Expectations and Assumption

05 Failure to Declare, Track & Reduce Risk & Uncertainties

06 Lack of Internal Peer Review

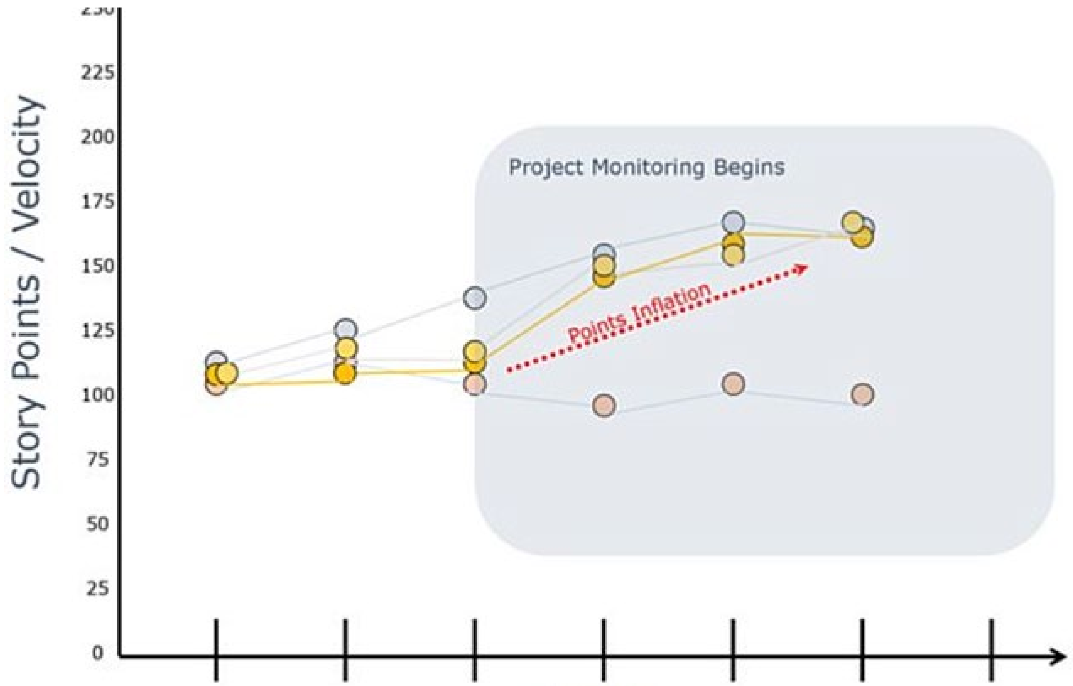
07 Lack of Estimation Experience

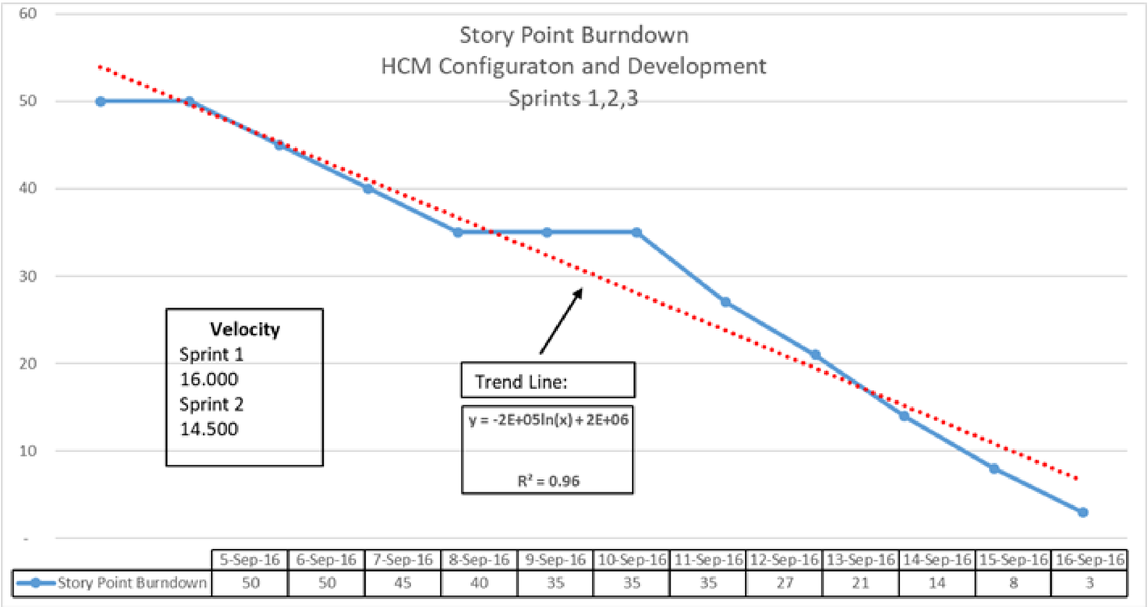
08 Failure to Consider Environmental Factors

09 Failure in the Estimation Tool/Process

10 Estimating to a Target Assumption

MANAGING MODERN SOFTWARE DEVELOPMENT PROGRAMS





Story Point Burndown
HCM Configuraton and Development
Sprints 1,2,3

Velocity
Sprint 1
16.000
Sprint 2
14.500

Trend Line:

y = -2E+05ln(x) + 2E+06

$R^2 = 0.96$

	5-Sep-16	6-Sep-16	7-Sep-16	8-Sep-16	9-Sep-16	10-Sep-16	11-Sep-16	12-Sep-16	13-Sep-16	14-Sep-16	15-Sep-16	16-Sep-16
Story Point Burndown	50	50	45	40	35	35	35	27	21	14	8	3

Measure the Right Thing – Backlog – Velocity – Burndown Charts
Manage Expectations / Set Realistic Time Frames
Align the Work Streams
Seek Objectivity

KEY QUESTIONS TO ASSESS THE QUALITY OF THE AGILE PROGRESS

Size Metrics: “Are the user stories consistent and do they follow the basic structure of ‘As a__, I Want, So That…’?”

Agile Methods: “Is the development process using the proper Agile method for work to be performed?”

Agile Integrity: “Is the program deviating from the Twelve Agile Principles?”

Agile Delivery: The acquisition manager needs to recognize that Agile is a development mindset (not a methodology) created by practitioners trying to resolve the iron triangle of scope, schedule, and resources

Velocity: “Is velocity based on a historical baseline of the program?”

Governance: “Is there an experienced Scrum Master?”

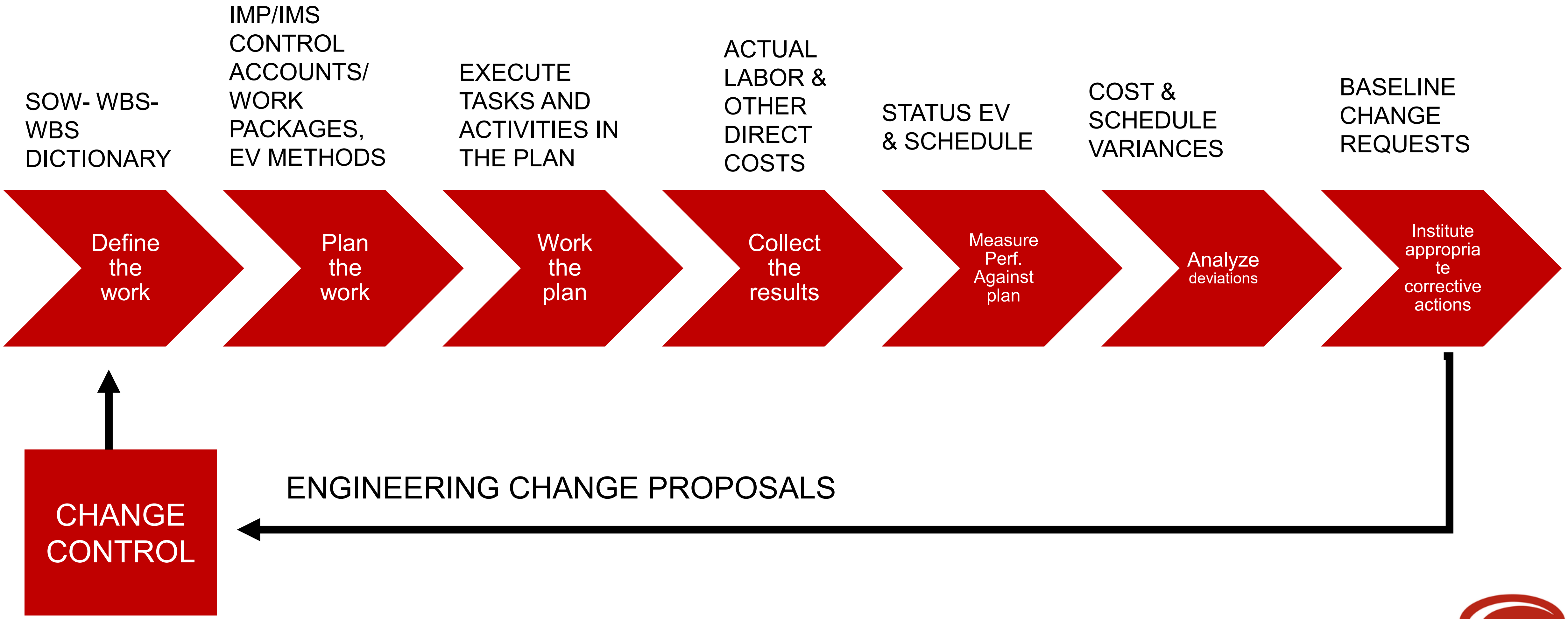
Expectations: “Is the team promising faster schedule and cheaper cost?” Research consistently shows

Agile Values: “Are the Agile values being embraced or is it simply the method du-jour?

GALORATH

EARNED VALUE MANAGEMENT PROCESS

Consider Tailored or EVM Lite Options

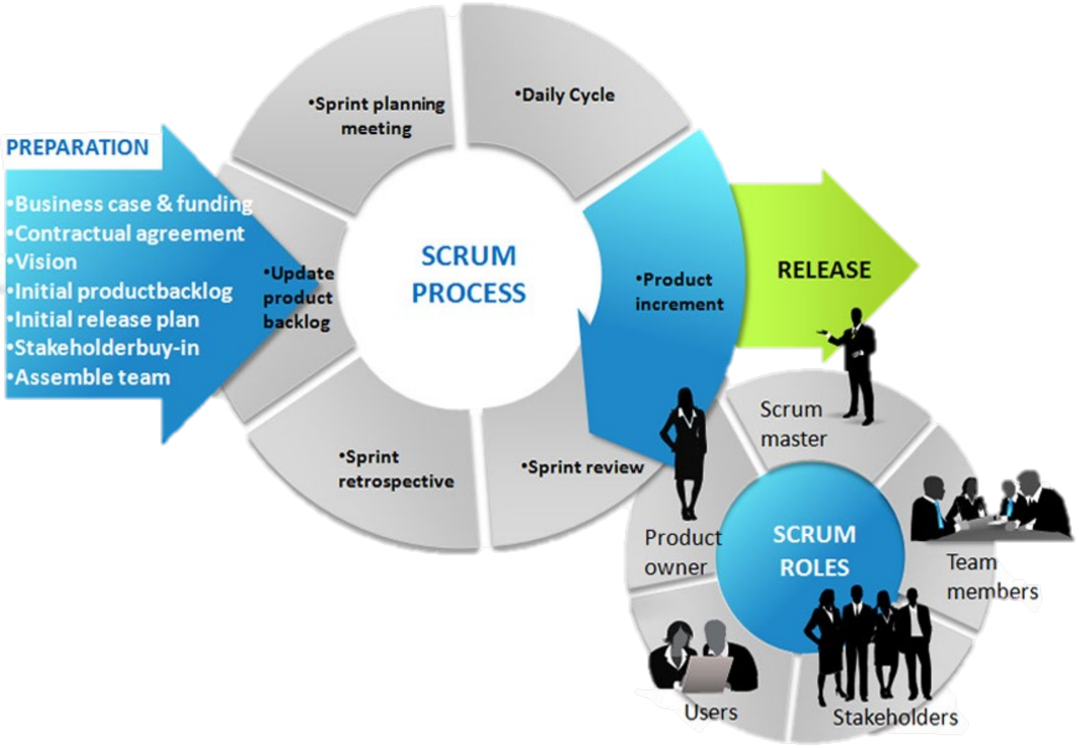

SOW- WBS- WBS DICTIONARY
IMP/IMS CONTROL ACCOUNTS/ WORK PACKAGES, EV METHODS
EXECUTE TASKS AND ACTIVITIES IN THE PLAN
ACTUAL LABOR & OTHER DIRECT COSTS
STATUS EV & SCHEDULE
COST & SCHEDULE VARIANCES
BASELINE CHANGE REQUESTS
Define the work
Plan the work
Work the plan
Collect the results
Measure Perf. Against plan
Analyze deviations
Institute appropria te corrective actions
ENGINEERING CHANGE PROPOSALS
CHANGE CONTROL

TYPICAL HYBRID AGILE DEVELOPMENT

Water-Scrum-Fall

REQUIREMENTS

DESIGN



SCRUM ROLES ARE CRITICAL

Testing and Sustainment (sometimes in the Sprint sometimes a separate activity)

TWO CLASSES OF FEDERAL PROGRAMS

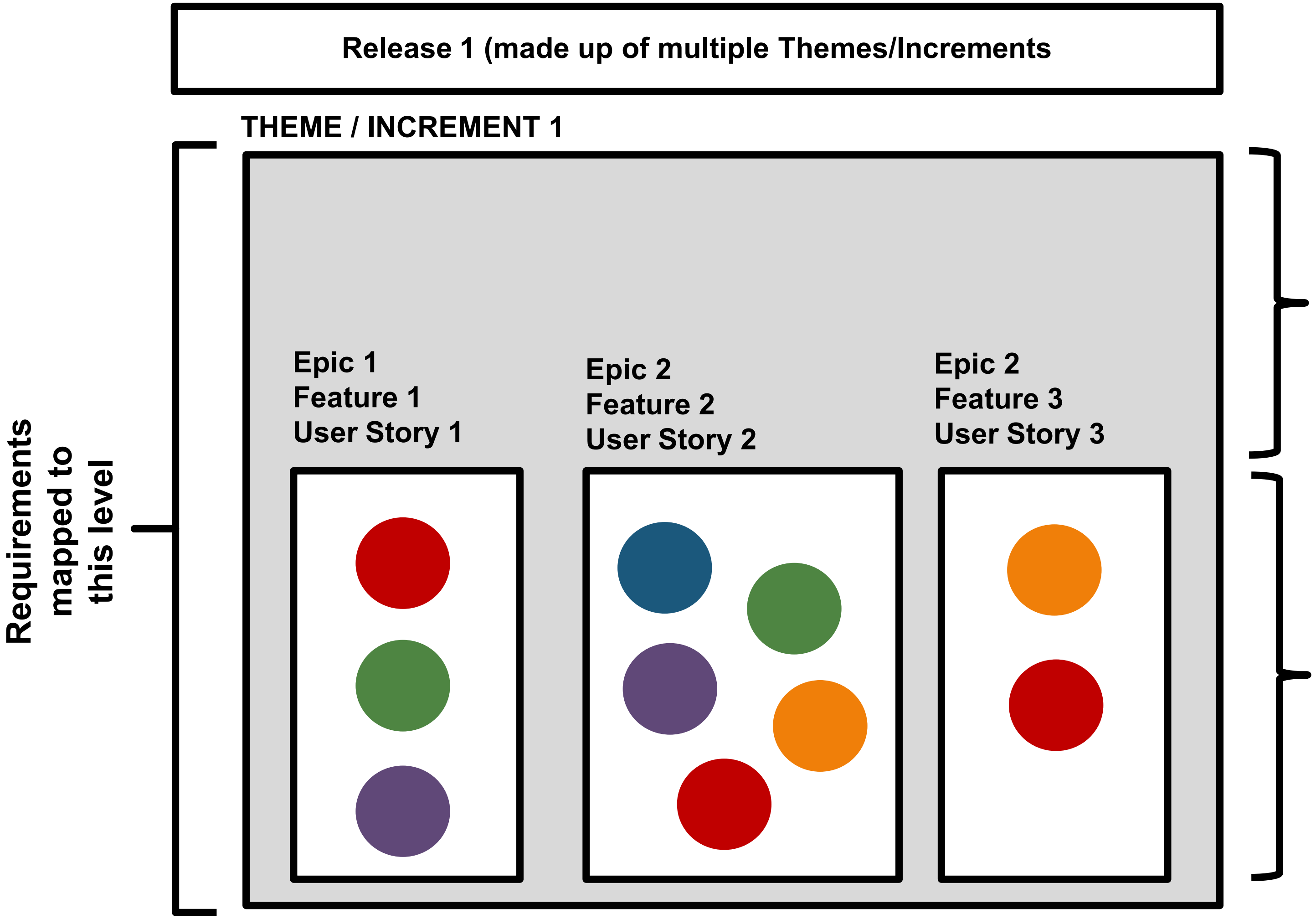
Incremental programs

- Follow the commercial Agile practices
  - Small user stories
  - Single sprint, or even multiple user stories being completed in a single sprint
- Generally not applying a full EVM process

Transformational programs

- Creating completely new capabilities
- “Hybrid-Agile” approach applied
  - Longer sprints
  - Larger conceptual stories/features
  - Full EVM process.

AGILE/EVM BUILDING BLOCKS*

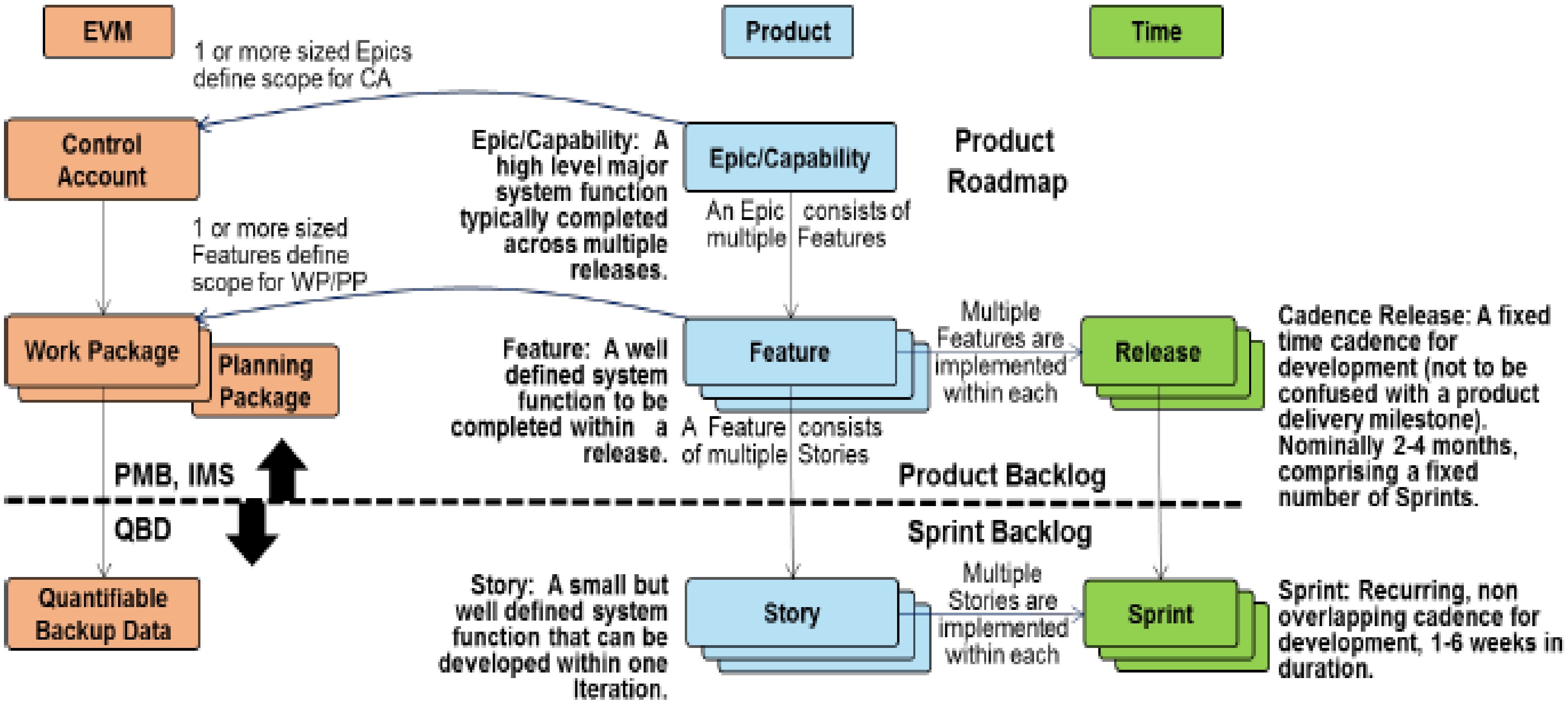


EVM **work Packages** identified at Epic or Theme level
Sprints may be too low a level to track as a work package

SPRINTS
Cost Estimating often done at the Sprint Level

* These "building blocks" are program specific and may be called by different names

AGILE- EVM RELATIONSHIPS


EVM
Product
Time
1 or more sized Epics define scope for CA
Control Account
Epic/Capability: A high level major system function typically completed across multiple releases.
Epic/Capability
Product Roadmap
An Epic consists of multiple Features
1 or more sized Features define scope for WP/PP
Work Package
Planning Package
Feature: A well defined system function to be completed within a release.
Feature
Multiple Features are implemented within each
Release
Cadence Release: A fixed time cadence for development (not to be confused with a product delivery milestone). Nominally 2-4 months, comprising a fixed number of Sprints.
A Feature consists of multiple Stories
PMB, IMS
QBD
Product Backlog
Sprint Backlog
Quantifiable Backup Data
Story: A small but well defined system function that can be developed within one Iteration.
Story
Multiple Stories are implemented within each
Sprint
Sprint: Recurring, non overlapping cadence for development, 1-6 weeks in duration.

NDIA “An Industry Practice Guide for Agile on Earned Value Management Programs” v1.1 March 31, 2017

WHERE DOES EVM FIT IN?

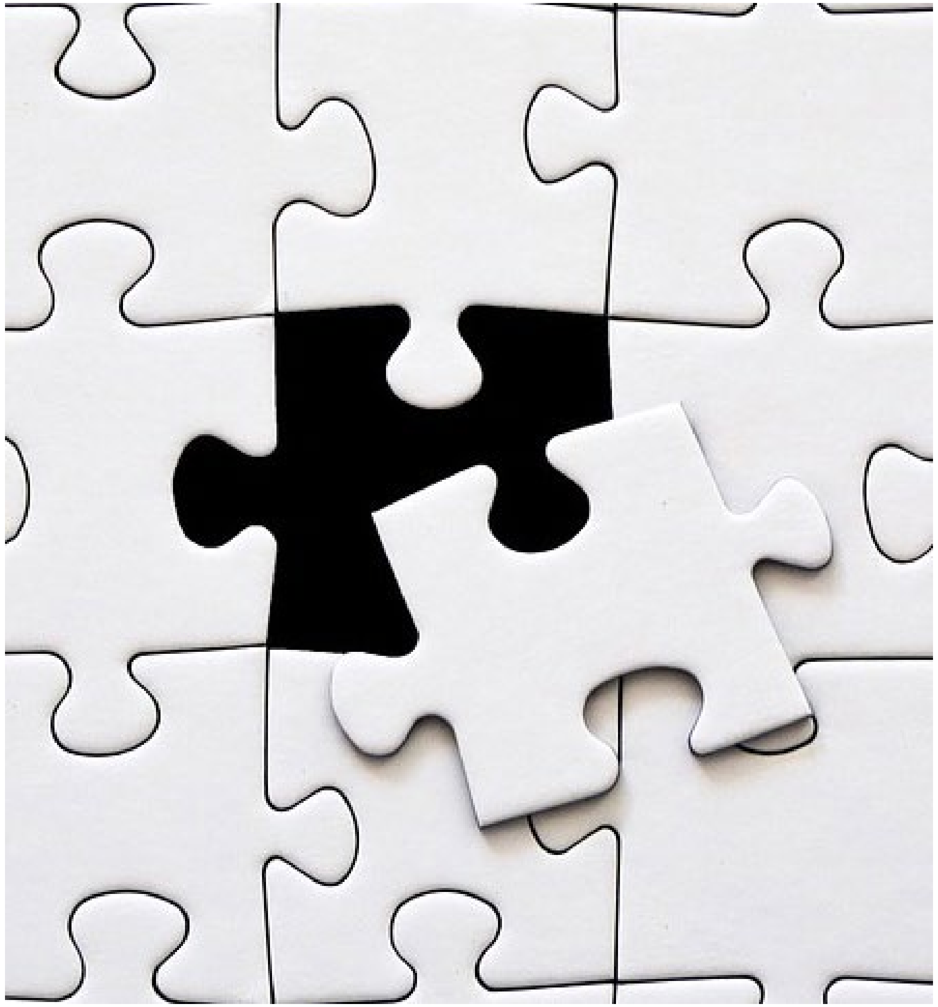
As long as there is a plan and product(s) EVM can be applied

The product backlog defines the product, and sprints are used to time phase the work.

Status of each Sprint is rolled up to a level, e.g. Control Accounts (CA)

- Below the EPIC is the Feature at the Work Package (WP) level, which breaks the EPIC into functional packages
- Features are decomposed into Stories and Story points
- Sprints are statused by, in this case, Stories and Story Points, which are maintained in an Agile Program Management Tool
- A feature may require multiple Spints to complete
- No credit given for a Story not completed in a Sprint (backlogged)

As Features are completed the percent complete is rolled up to the EPIC level



CONCLUSIONS

The adoption of Agile practices in federal programs is growing and changing the way we approach software

Modern federal software development programs are evolving into incremental development practices and some of them are starting to adopt the Agile development framework.

✓ Some of these programs are fully adopting the framework while some others are “executing Hybrid Agile. Regardless of level of Agile adoption, these federal programs can benefit from using a formal estimation and Earned Value Management (EVM) processes.

Robust cost baselines become the baseline from which performance is measured

Earned Value Management can be applied to Agile

Federal programs need to adopt continuous iterative development best practices for software

FOR MORE INFORMATION

REACH OUT

BOB HUNT

President
Galorath Federal
Bhunt@Galorath.com

DAN GALORATH

CEO Galorath
Incorporated;
Galorath@Galorath.com

IAN BROWN

Director of Operations &
Systems Analysis
Galorath Incorporated
Ibrown@Galorath.com

DAVID DEWITT

Director Commercial and
International Program
Galorath Incorporated
ddewitt@galorath.com

ESTEBAN SANCHEZ

Senior Cost Estimation
Consultant
Galorath Incorporated
esanchez@galorath.com

GALORATH.COM