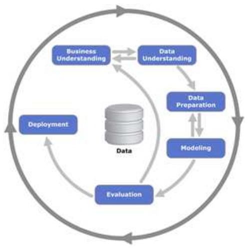
Waterfall
vs
Agile

(Assembly Methods)

PMI® Project Management Institute

Scaled Agile® Framework

Richard Frederick, PMP (rfrederick.pmp@gmail.com; 214-755-7035 talk or text; www.linkedin.com/in/rfrederick)



CBDA CERTIFIED IIBA®

ISO International Organization for Standardization

ANSI American National Standards Institute

sas®

OMG® OBJECT MANAGEMENT GROUP®

INCOSE International Council on Systems Engineering

IEEE®

Requirements Development Management (RDM) for

Business Data Analytics

(The Language of Data)

PMI® Project Management Institute

Scaled Agile® Framework

IIBA® International Institute of Business Analysis

UNIFIED MODELING LANGUAGE UML™

W3COMPUTING A DEVELOPER REFERENCE WEBSITE

Requirements Agenda

Business Data Analytics "The Language of Data"

01-FOUNDATION

- **LAB-Introductions (Name, Job Title, Objectives)**
- Dashboards and Robots (Data Mining & Machine Learning)
- Requirements and Testing (Four Quadrants)
- What are Structured Language Requirements? (Structured English and Structured Query Language)
- Why Should You Care? (Primary Source of Project Problems)
- How Do They Work? (Discreet Intellectual Property Inventory)
- Types of Requirements (Product, Project, DATA)
- Natural Language Processing (Morphology, Semantics, Syntax and Linguistics)
- OMG-SBVR (Semantics of Business Vocabulary & Rules)
- IEEE-EARS (Easy Approach to Requirements Syntax)
- INCOSE (Rules for Writing Requirements) & QVscribe
- Waterfall and Agile (Assembly Methods)

02-ELICIT

- **LAB-Vision/Scope (Seek to Understand)**

Elicitation Techniques:

- Document Analysis (Low Hanging Fruit)
- Interface Analysis (Navigation & Functionality)
- Benchmarking (Actual Data)
- Brainstorming (Every Idea is a Good Idea until it becomes a Bad Idea)
- Prototyping (Minimum Viable Product)
- Reverse Engineering (Begin with the End in Mind)
- Interview (Thinking Questions)
- Workshop (Group Interviews)
- Observation (What do you See?)
- Survey Questionnaire (Paper equals proof)

03-ANALYZE

- What are Models? (Pictures of Language)
- **LAB-The Language of Modeling (GIVEN pre WHEN process THEN output-result)**
- Types of Models (Context-Structure, Usage, Data Behavior, Process Flow)
- Context-Structure (Vision, Roadmap, Scope WBS)
- Usage (EPIC, UseCase, UserStory, Feature)
- Data Behavior (ERD, JOIN-Denormalization, Star Schema, Dimensional OLAP, Dashboard, Intelligence)
- Data Behavior (Data Dictionary, DataFlow, Data Structure Instance, Data Element Attribute, Data Store)
- Data Behavior (Process Logic, Business Rules)
- Process Flow (Swimlane)

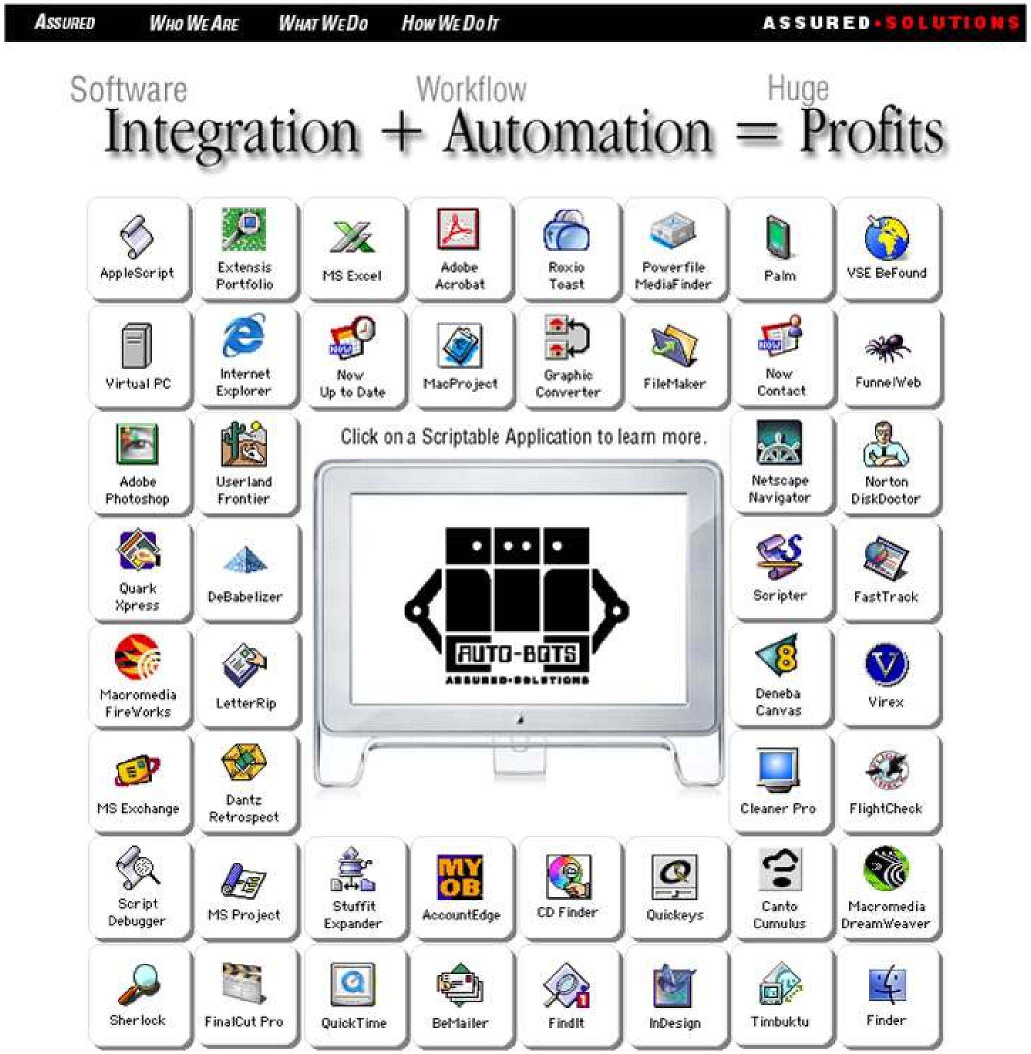
04-DOCUMENT

- Categorization, Organization, Documentation, Integration, Automation
- Making Documents Easy to Read (Fonts & Navigation)
- Document Types (BRD, TRD)
- **LAB-Business Requirement Document (Concept of Operation)**
- Technical Requirement Document (System Specification)

05-VALIDATE

- Validation thru Triangulation (Prep Drills)
- Traceability (Project Unique Identifier)
- Requirements Baseline (ROM Estimate, Planning Estimate, Definitive Estimate)
- **LAB-Estimating Story Points (Complexity and Risk)**
- Lessons Learned (Course Wrap-Up)

Assured Solutions



Software Integration + Workflow Automation = Huge Profits

Click on a Scriptable Application to learn more.

Overview “Seek to Understand”

- Software Assembly
  Analyze > Design > Code > Test
- Waterfall Assembly
  “Scope drives Schedule”
- Agile Assembly
  “Schedule drives Scope”
- Agile Planning
  Roadmap > Backlog > Release

Waterfall
vs
Agile

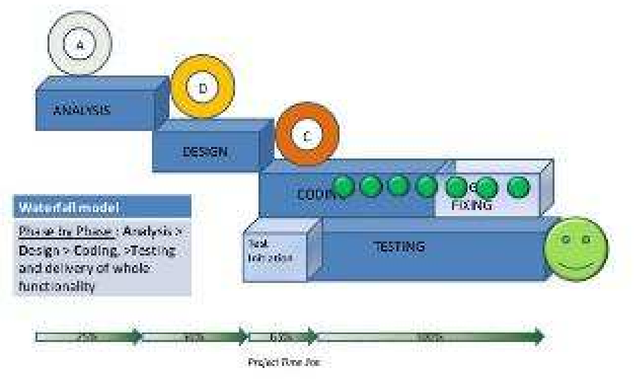
(Assembly Methods)

PMI® Project Management Institute

Scaled Agile® Framework

Software Assembly Methods (Analyze > Design > Code > Test)

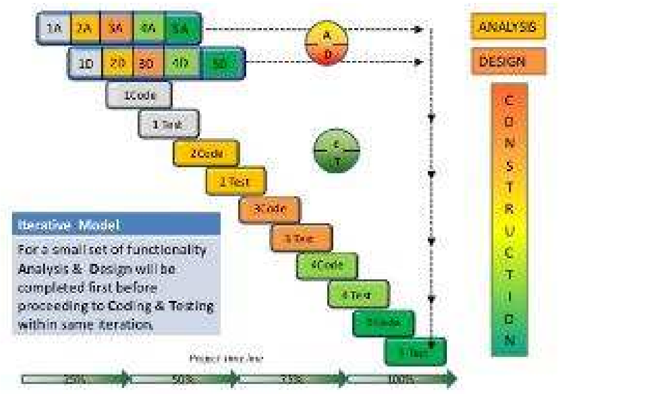
WATERFALL Model



Waterfall model

Phase by Phase : Analysis > Design > Coding, >Testing and delivery of whole functionality

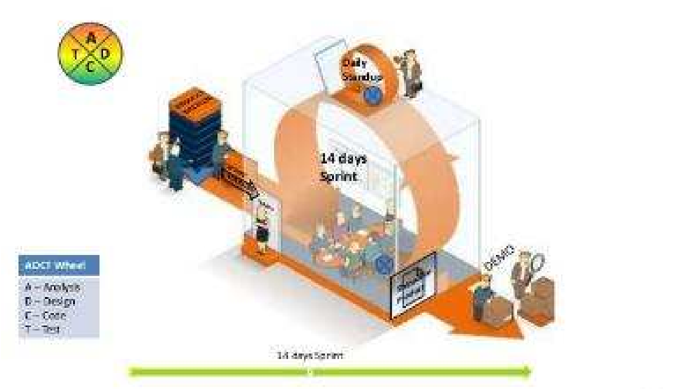
ITERATIVE Model



Iterative Model

For a small set of functionality Analysis & Design will be completed first before proceeding to Coding & Testing within same iteration.

SCRUM - Sprint



ADCT Wheel

A – Analysis
D – Design
C – Code
T – Test

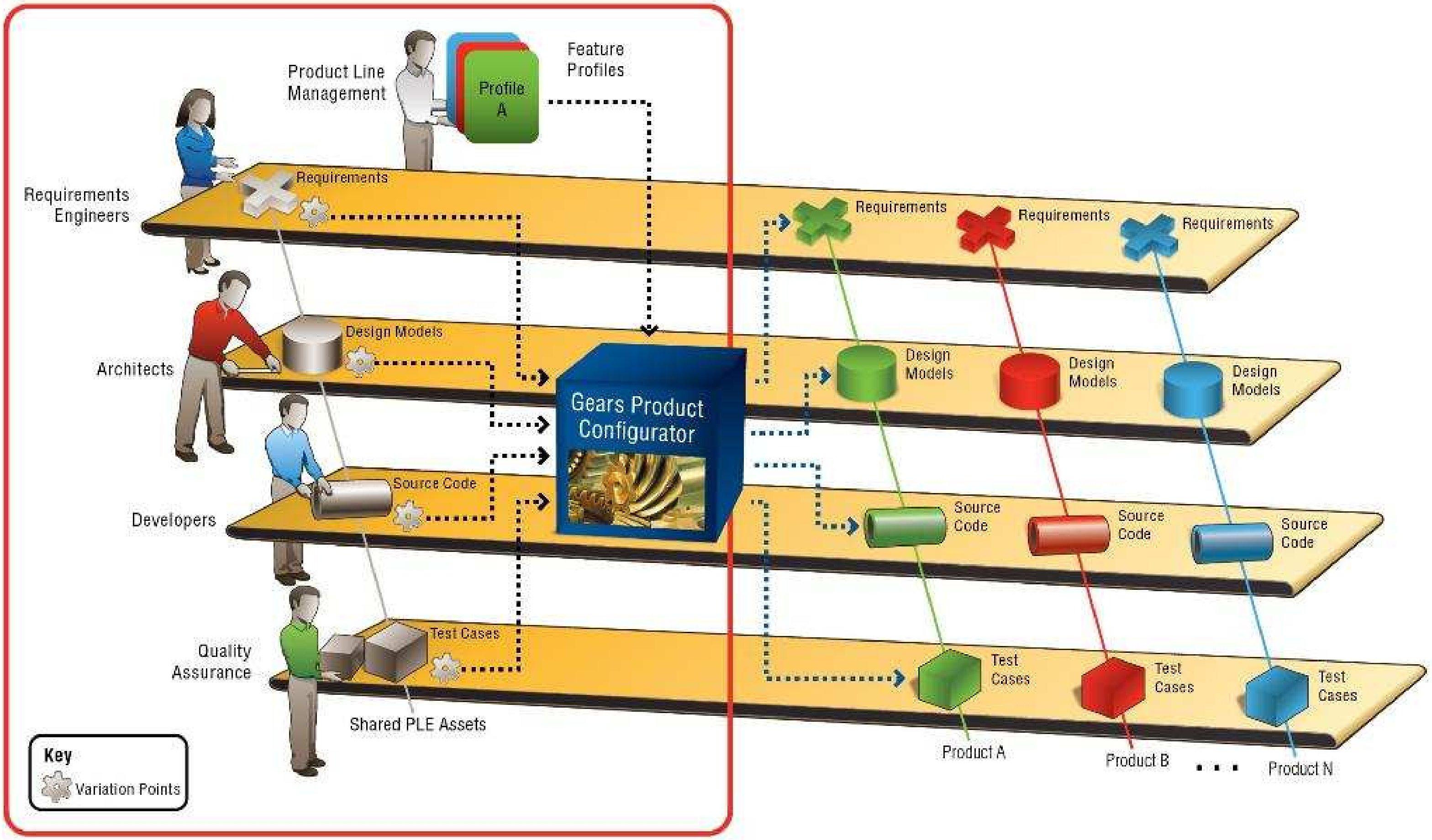
Software Assembly Methods (Analyze > Design > Code > Test)

Analyze

Design

Code

Test



Understanding Assembly

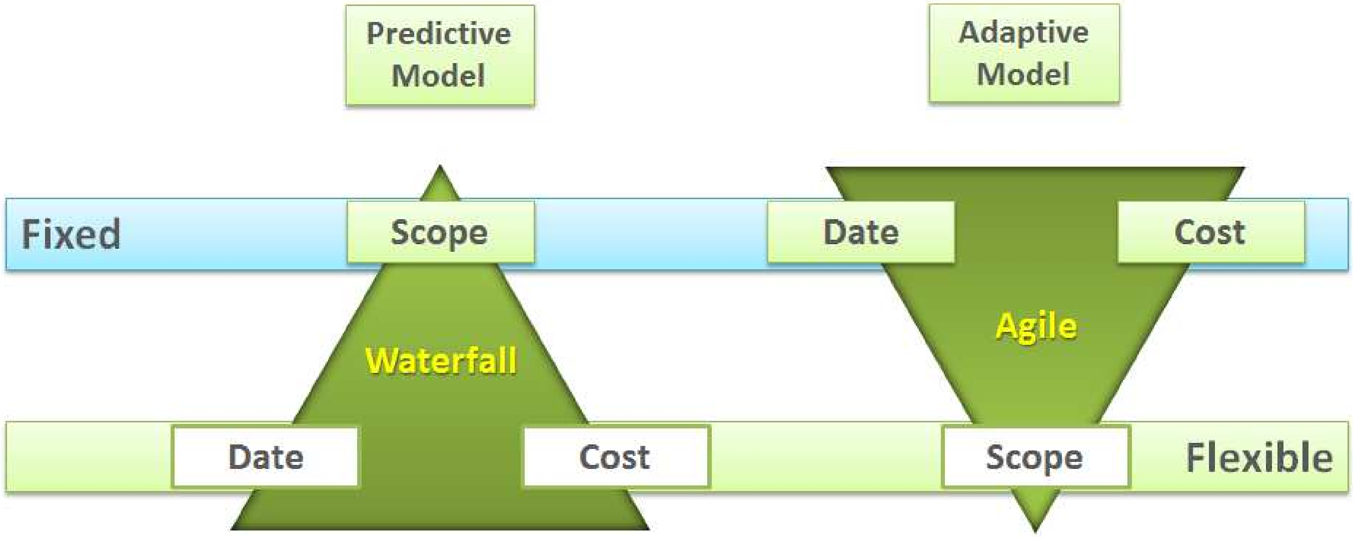
- Assembling "things" requires understanding four key concepts:
  1. What are the parts you are assembling? This is SCOPE
  2. How long will it take to assemble them? This is SCHEDULE
  3. How complex is it? This is RISK
  4. How much will you spend? This is COST

Waterfall Assembly vs Agile Assembly

Waterfall Assembly	**Agile Assembly**
SCOPE drives SCHEDULE	SCHEDULE drives SCOPE
Lockdown Scope, then assemble Transactions along a "Critical Path" Schedule	Lockdown Schedule, then determine "How Much Scope (called Story Points)" will fit within a two week "sprint" of 80 business hours.
Risks are measured against the overall project Scope and adjustments to Schedule are added as Contingency Reserve.	Risks are measured against Scope to identify "Complexity" and calculate "Story Point" values.
Work is Assigned "Top Down"	Work is Selected "Bottom Up"

Understanding Assembly

Waterfall (Predictive) vs Agile (Adaptive)



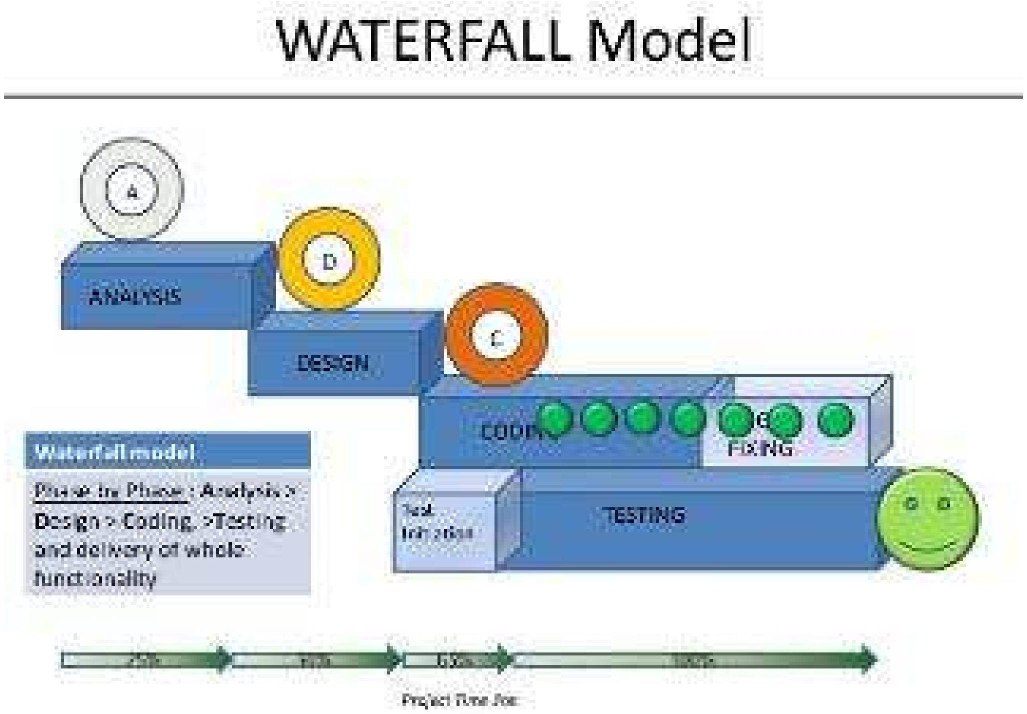
DSDM.org, 1994

Waterfall ASSEMBLY (Scope drives Schedule)

Waterfall Assembly

Scope drives Schedule

- Waterfall Assembly has been around for a very long time. It is the assembly method that most everyone uses to **build physical, tangible assets.**
  - With Waterfall, you break down a product into "discrete" individual tangible assets called Work Packages.
  - The idea to figure out exactly what you are going to assemble in the beginning, and then try to "lock-down" the deliverables into a SCOPE Baseline with minimal changes.
- From the SCOPE Baseline, you create the list of assembly tasks and activities (which I call "discrete" transactions for accounting purposes) and arrange them along a SCHEDULE timeline called a "Critical Path."
  - Once you have a SCHEDULE timeline, then you analyze how complex or "risky" the work is to complete.
  - This RISK Analysis is then added as a Contingency Reserve to your project to compensate for any "Surprises."
- Finally, knowing all of this up front allows you to develop a COST Baseline to help track spending.

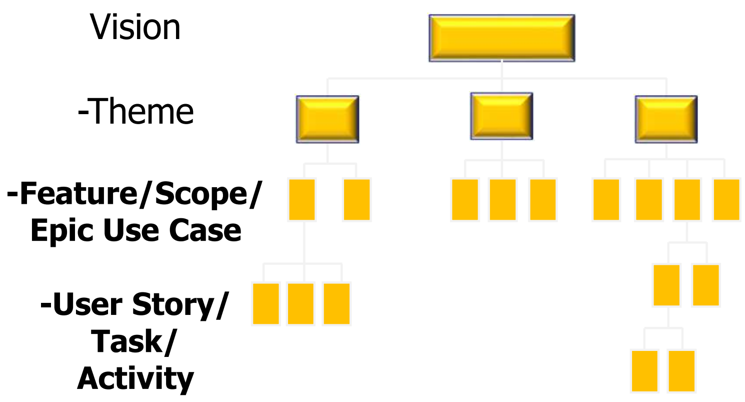


AGILE PLANNING PROCESS = ROADMAP > BACKLOG > RELEASE

Agile Assembly

Schedule drives Scope

1. ROADMAP (Structured)



Vision

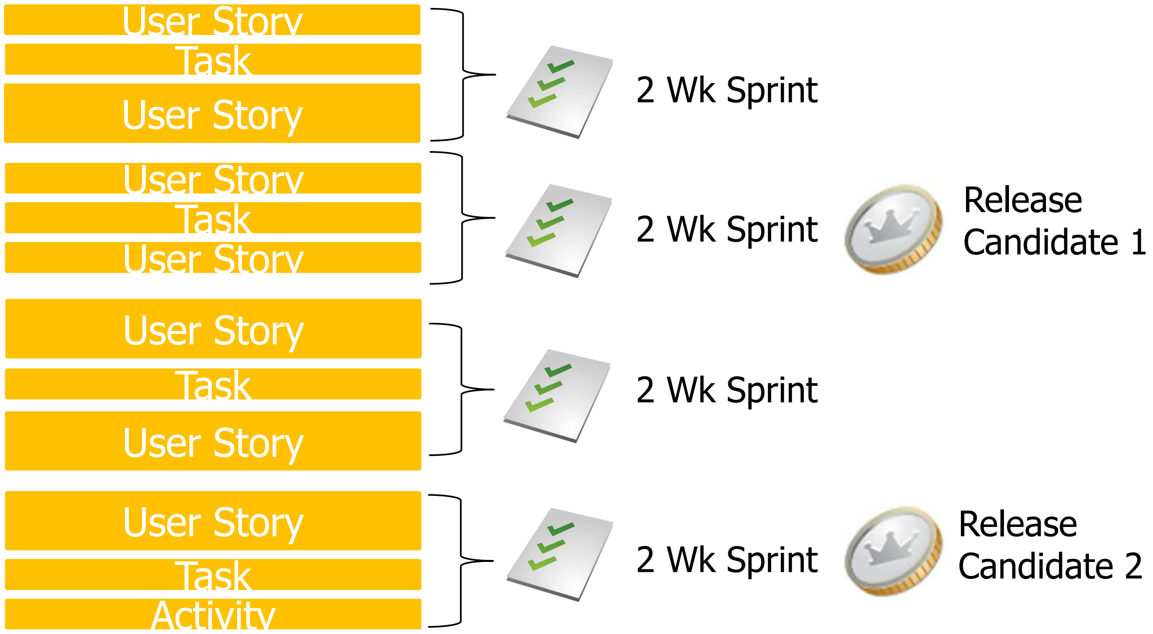
-Theme

-Feature/Scope/ Epic Use Case

-User Story/ Task/ Activity

2. BACKLOG

3. RELEASE



User Story
Task
User Story
User Story
Task
User Story
User Story
Task
User Story
User Story
Task
Activity

2 Wk Sprint

2 Wk Sprint

Release Candidate 1

2 Wk Sprint

2 Wk Sprint

Release Candidate 2

- ✓ Feature/Scope comprised of User Stories of various **SIZES (aka Story Points)**
- ✓ Feature/Scope has **Business VALUE** based on Minimized Costs or Maximized Revenues
- ✓ Feature/Scope has **different PRIORITIES** that must be defined

User Stories and Tasks are placed on the BACKLOG in Order of **SIZE, VALUE, PRIORITY**

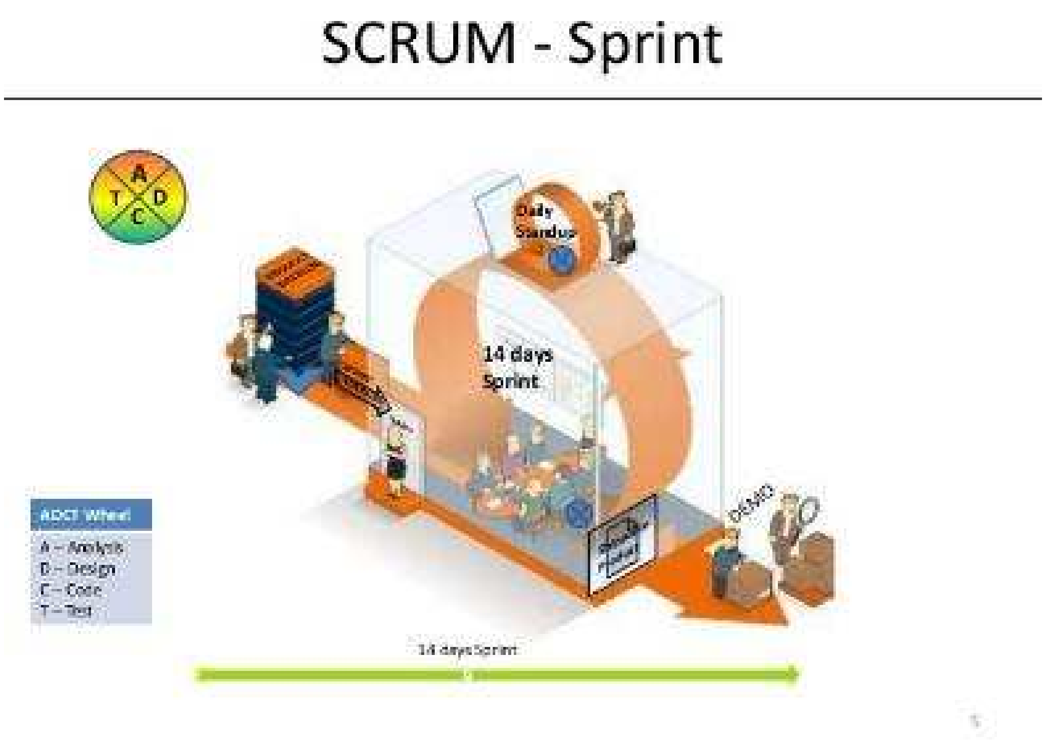
At RELEASE PLANNING, we decide when our PRODUCT will meet Acceptance Criteria and provide **Business VALUE** based **upon Reduced Costs, Increased Revenue or other KPIs.**

Agile ASSEMBLY (Schedule drives Scope)

Agile Assembly

Schedule drives Scope

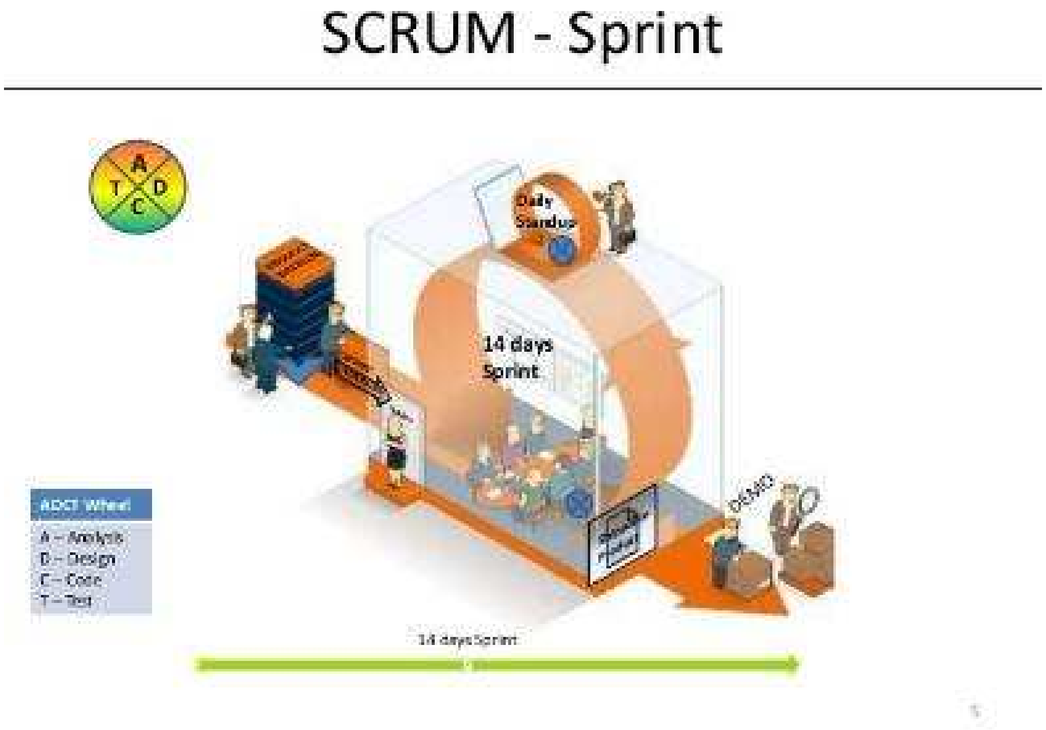
- Agile Assembly is a relatively recent and popular method for assembling **non-tangible, intellectual property assets** like software.
  - With Agile, your SCHEDULE is "locked-down" to a specified amount of time, usually two weeks in length, called a Sprint (Iteration).
  - The idea behind Agile is to answer the question: "How much Work (SCOPE) can you assemble during an 80hr Sprint (Iteration)?"
- Like Waterfall, the Work (SCOPE) is broken down into the smallest pieces (discrete assets and transactions) for estimating.
  - However, unlike Waterfall, where the amount of Work (SCOPE) determines the SCHEDULE, Agile estimates the amount of Work (SCOPE) based upon how Complex (RISK) it is to assemble.
- The Complexity (RISK) of the work is identified using a subjective measurement called “Story Points.”
  - “Story Points” are represented using a sequence of numbers: 1, 3, 5, 8, 13, 20, 40 and 100 to estimate Complexity (RISK). The greater the Complexity (RISK), the higher the Story Point number.
- Story Point “Complexity” is determined by team members who break down the project into small pieces (discrete components) and then "vote" on each component's Complexity (RISK).



Agile Assembly

Schedule drives Scope

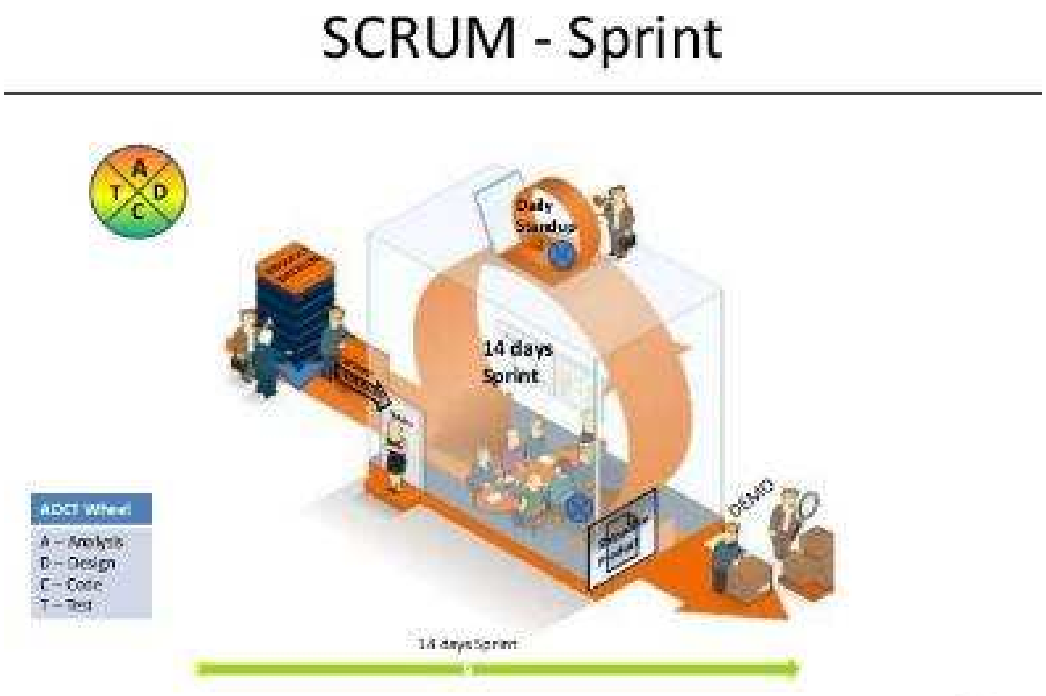
- For example, a discrete component of work with a story point of 40 is far more complex to assemble than a discrete component of work with a story point of 1.
  - Once the team agrees to a Story Point, that number is assigned to that discrete component of work.
  - This allows the team to determine which work should be assembled within a Sprint (Iteration).
- The total amount of Work (SCOPE) a team can assemble during a Sprint (Iteration) is called Velocity.
  - For example, one team might have a Velocity of 100 Story Points while another team might have a Velocity of 150 Story points.
  - The higher the velocity, the more work the team can assemble.
- Knowing the Estimated Velocity of each team allows the Product Owner (Business Analyst) and Scrum Master (Project Manager) to prioritize and SCHEDULE the Work (SCOPE).
- Finally, you can use a "time and materials" accounting approach to develop a COST Baseline to track spending.



Simplified SPRINT VELOCITY

Simplified Sprint Velocity

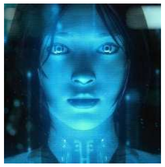
1. Two Week Sprint = 10 Business Days
2. 5 person team can complete 5 Story Points (SP) per Person Day
3. 5 SP x 5 People x 10 Days = VELOCITY of 250 SP for the Sprint
    - ➢ 1 SP = approx. 1.5 hrs.
    - ➢ 5 SP = approx. 7.5 hrs.
4. 52 Week Calendar Year = 26 Sprints



Summary “Train the Trainer”

Summary “Train the Trainer”

- Software Assembly
  Analyze > Design > Code > Test”
- Waterfall Assembly
  “Scope drives Schedule”
- Agile Assembly
  “Schedule drives Scope”
- Agile Planning
  Roadmap > Backlog > Release



Let’s stay in contact with each other…

Let’s stay in contact:

Richard Frederick, PMP

214-755-7035 (text or talk)

Rfrederick.pmp@gmail.com

www.meetup.com/tampa-bay-IIBA/

www.linkedin.com/in/rfrederick

meetings.hubspot.com/rfrederick-pmp