TSP-Agile Showdown

Alan Padula

Team Software Process Symposium
Phoenix, AZ
September 22-24, 2008

Agenda

- What are we solving for?
- What is the resistance to Agile? to TSP?
- How are we adapting?
  - Iterative TSP
  - TSP Plus
  - TSP-Agile Blend (TAB) Model
    - Iteration 0
    - Release Launch
      - Roles
      - Story Point Estimation/Agile Processes
      - Release Planning
      - Quality "Doneness"
    - Iteration Planning
- Summary

What is Intuit solving for?

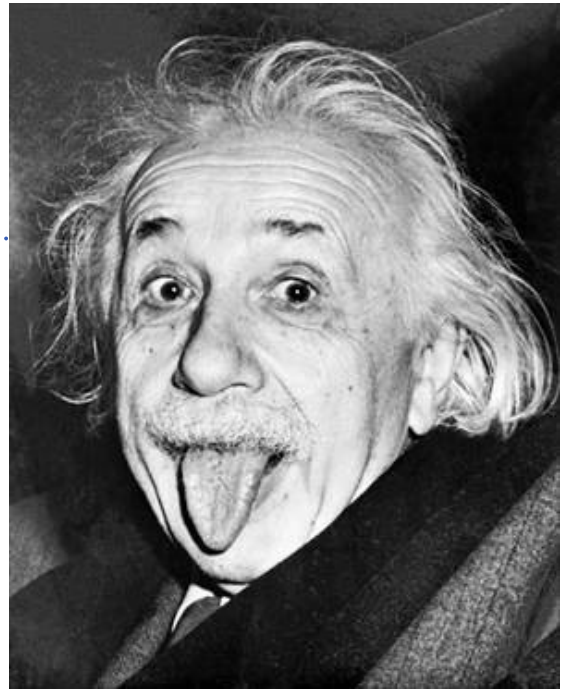
A Software Development Lifecycle that leverages

the best of the TSP and Agile development models

in order to create high-quality products
in a predictable and repeatable fashion

including projects with
rapidly changing or
unknown customer or
technology requirements.

What is the resistance to Agile? to TSP?



"Two things are infinite: the universe and human stupidity; and I'm not sure about the universe."

- Albert Einstein

Perception is the primary resistance to using either TSP or Agile

- **Perception of TSP**
  by an extreme "Agilist"
  - Heavyweight, overly burdensome, punitive process that cramps my creativity and productivity

- **Perception of Agile**
  by an extreme "TSP-er" or "Waterfall-er"
  - Ad-hoc, chaotic, shoot-from-the-hip low-quality producing brouhaha with no end-date or scope commitment

Agile Values

Perception of some TSP'ers

- Agile development values:
  - **Individuals & Interactions**
    over processes and tools
  - **Working Software**
    over comprehensive documentation
  - **Customer Collaboration**
    over contract negotiation
  - **Responding to Change**
    over following a plan

No processes!!!

No documentation!!!

No plans!!!

No quality!!!

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

intuit

TSP's Values
Managing by Data

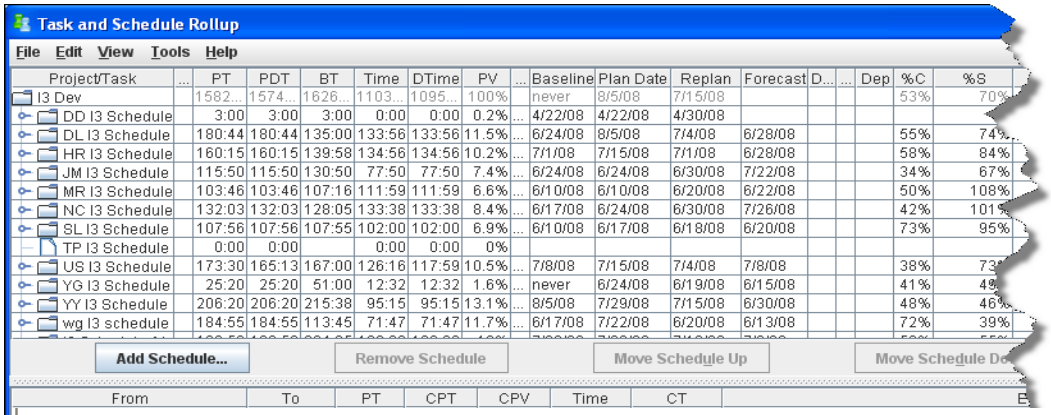
Perception of some Agilists

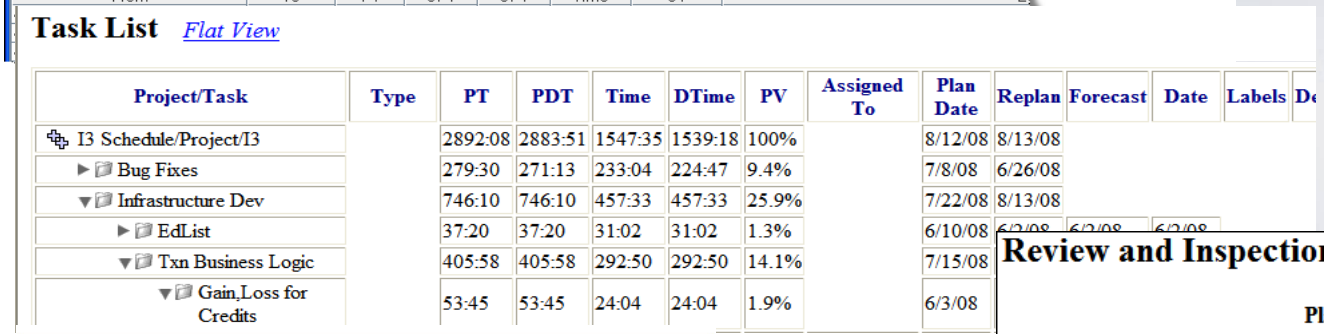
Too much process!!!

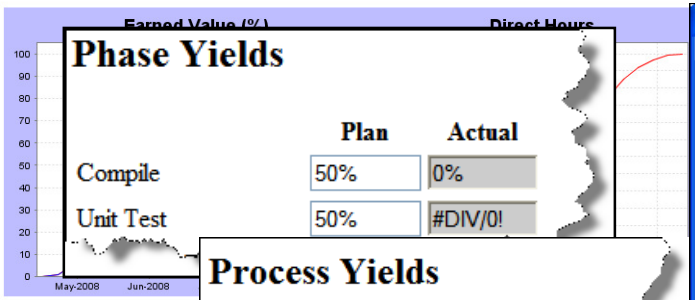
Too many metrics!!!

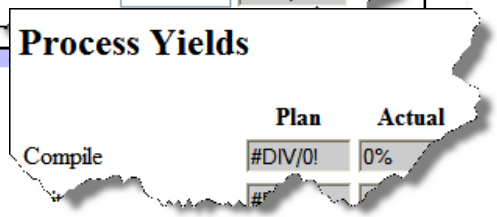
Cramps my style!!!

Will be used against me!!!



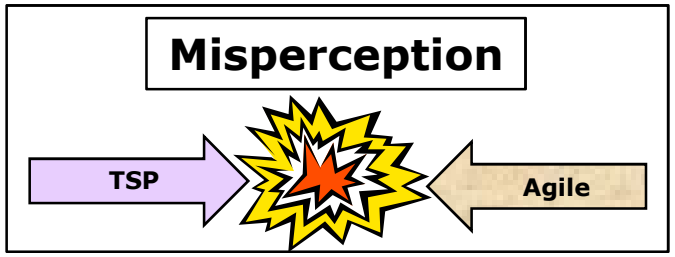


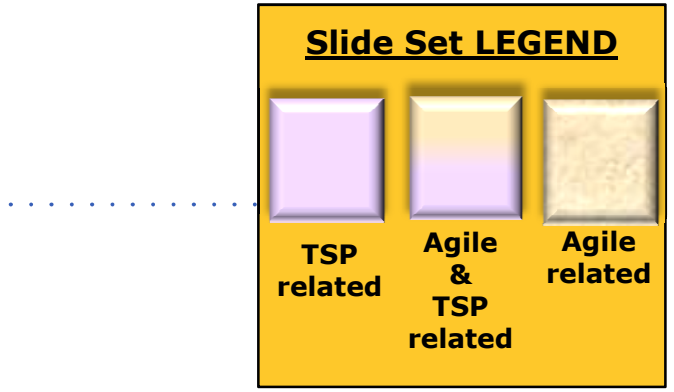


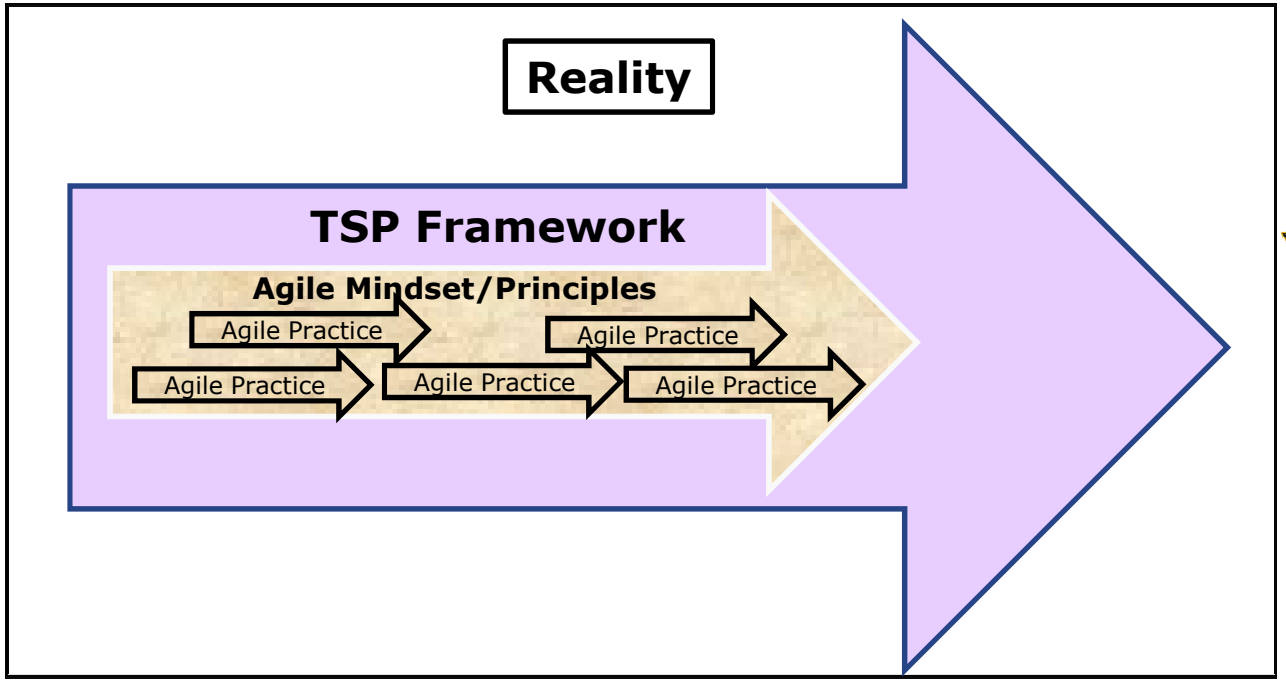


intuit

Perception & Reality





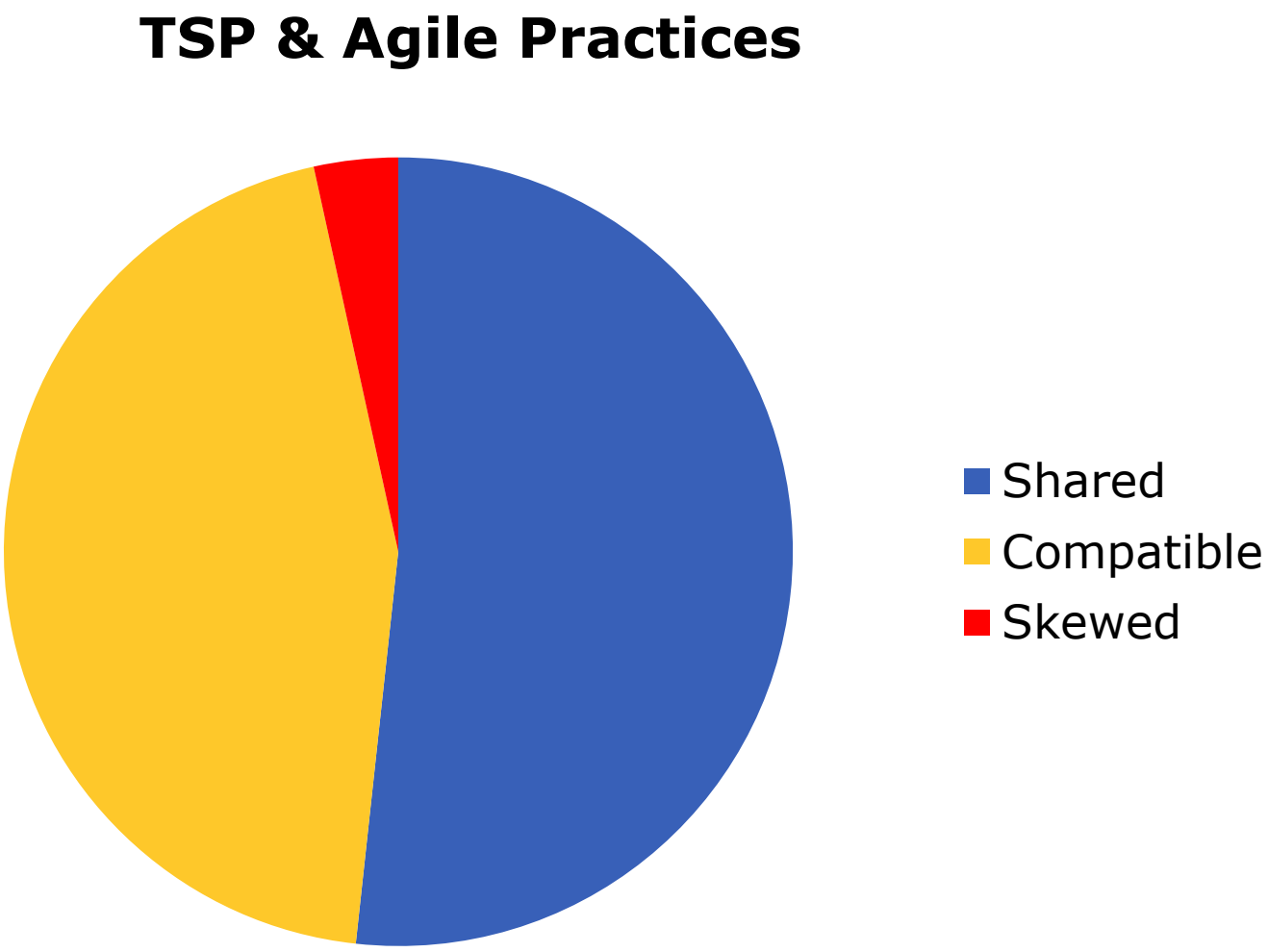


"Play Well Together!"

TSP & Agile are much more closely aligned than many TSP or Agile advocates might think!

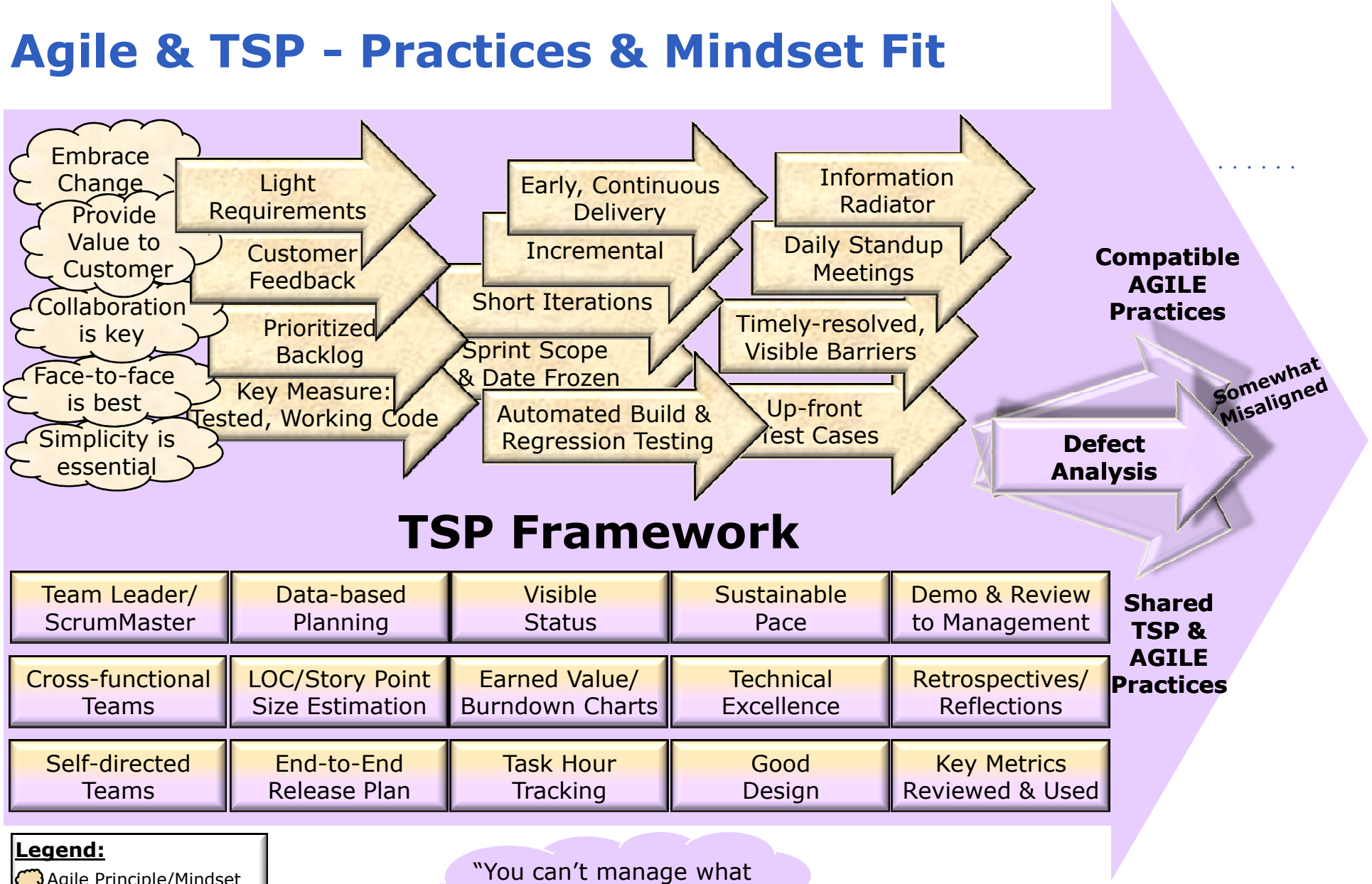
intuit

Agile & TSP Are Very, Very Aligned*



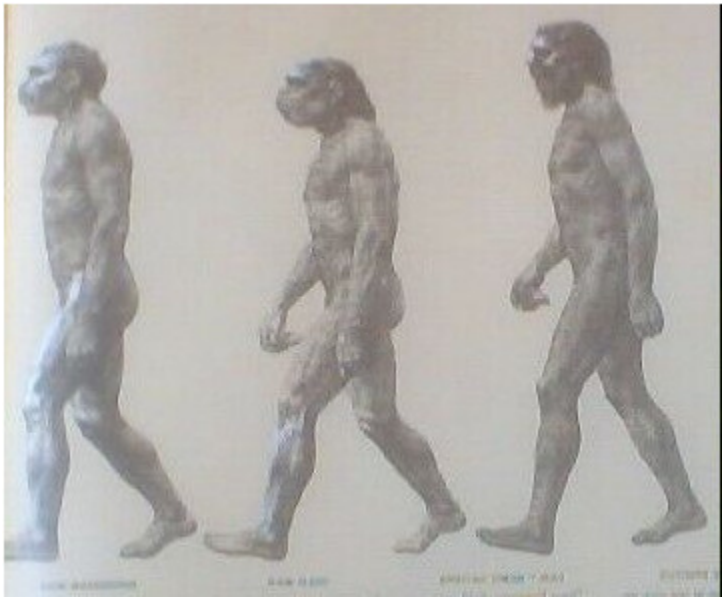
** This is just to make a point! There is no actual scientific or statistical data behind this.*

Agile & TSP - Practices & Mindset Fit



intuit

How are we adapting?



"It is not the strongest of the species that survives, nor the most intelligent that survives. It is the one that is most adaptable to change."

- Charles Darwin

intuit

Agenda

- What are we solving for?
- What is the resistance to Agile? to TSP?
- How are we adapting?
  - Iterative TSP
  - TSP Plus
  - TSP-Agile Blend (TAB) Model
    - Iteration 0
    - Release Launch
      - Roles
      - Story Point Estimation/Agile Processes
      - Release Planning
      - Quality "Doneness"
    - Iteration Planning
- Summary

You are here!

Alternative 1: Iterative TSP

Option 1: Fully-estimated conceptual design; content of all iterations defined

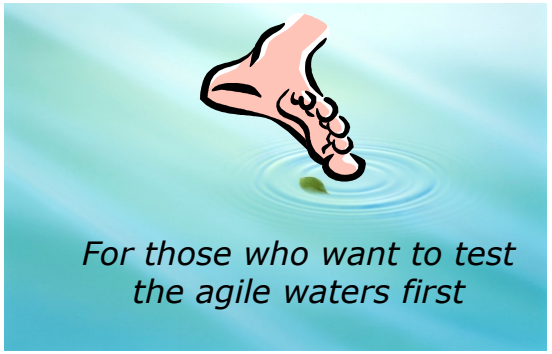
Option 2: Fully-estimated conceptual design; only current & next iteration content defined

Especially good for projects with:

- Well-known requirements
- Well-understood technology
- Little need to get feedback from the customer and want
- Greater visibility into planned progress against well-defined milestones, and
- Deep insights into their quality levels

Alternative 2: TSP Plus

Iterative TSP Frontend with an Agile Backend



Starting with..

TSP Launch --- Execute - Track - Replan --- Review

Rinse & Repeat

Especially good for projects that:

- **Meet the same criteria as "Iterative TSP"**
- **AND, want to try out or take advantage of a few, select Agile practices**

Common Intuit Agile Execution Elements

- **Daily Standups (2+/week)**
- **TDD or Concurrent Test Dev**
- **Demonstrable deliverables**
- **Customer feedback integration**
- **And many less common ones**

intuit

EMERGING & EVOLVING

Alternative 3:

TSP-Agile Blend

USA Swim Team

400m Medley

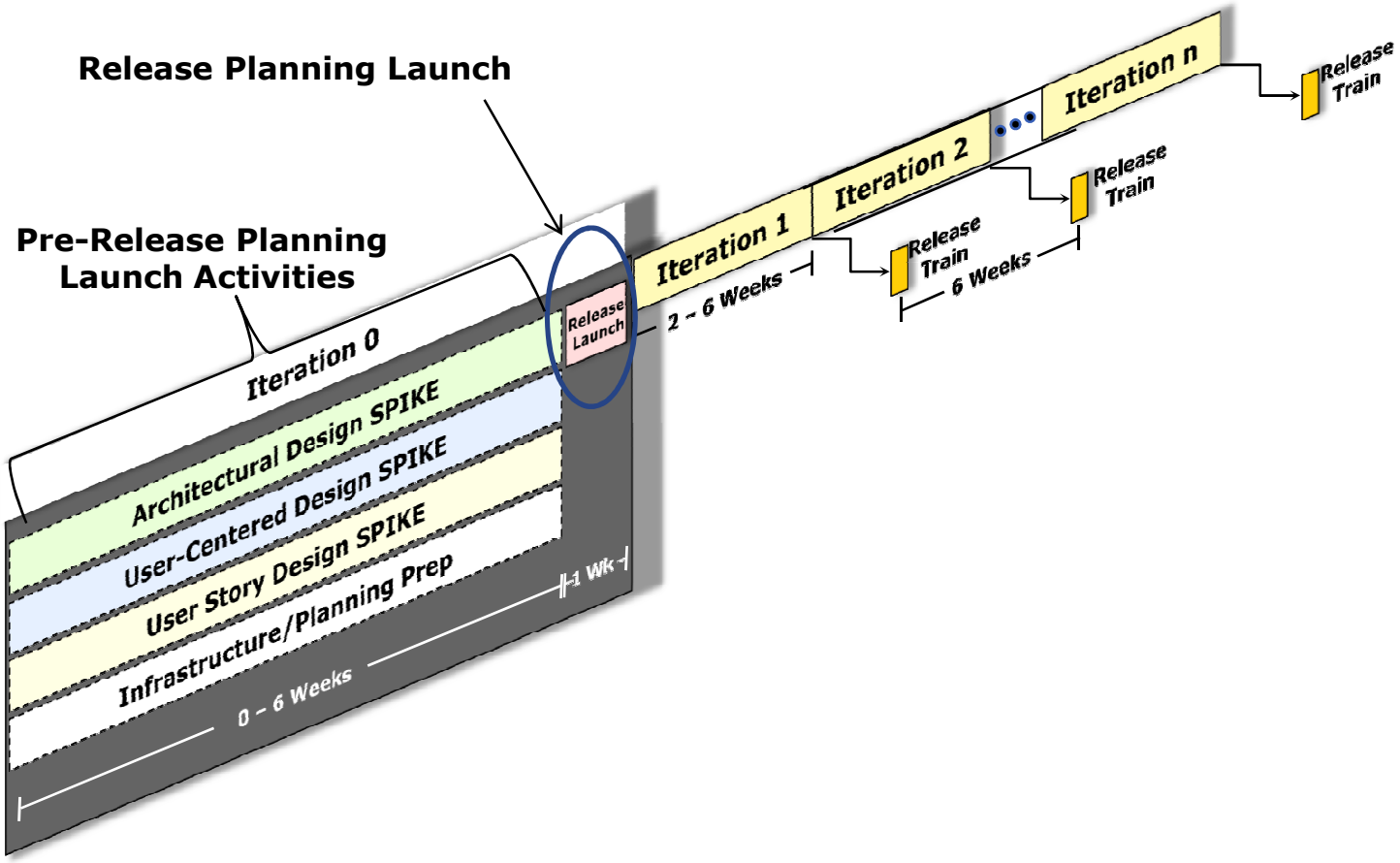
medley

Noun

1. a mixture of various elements

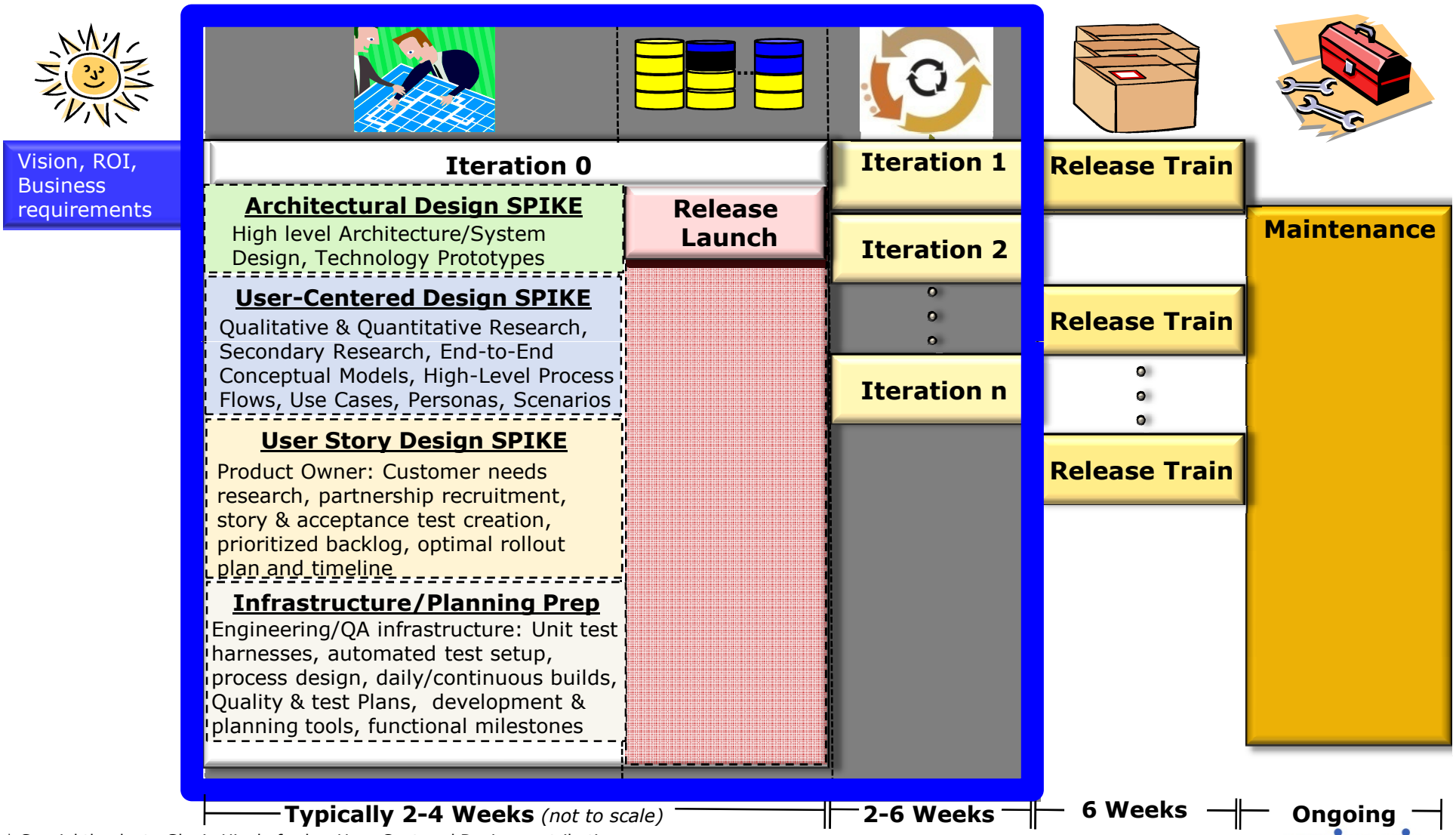
intuit.

TSP Agile-Blend (TAB) Model



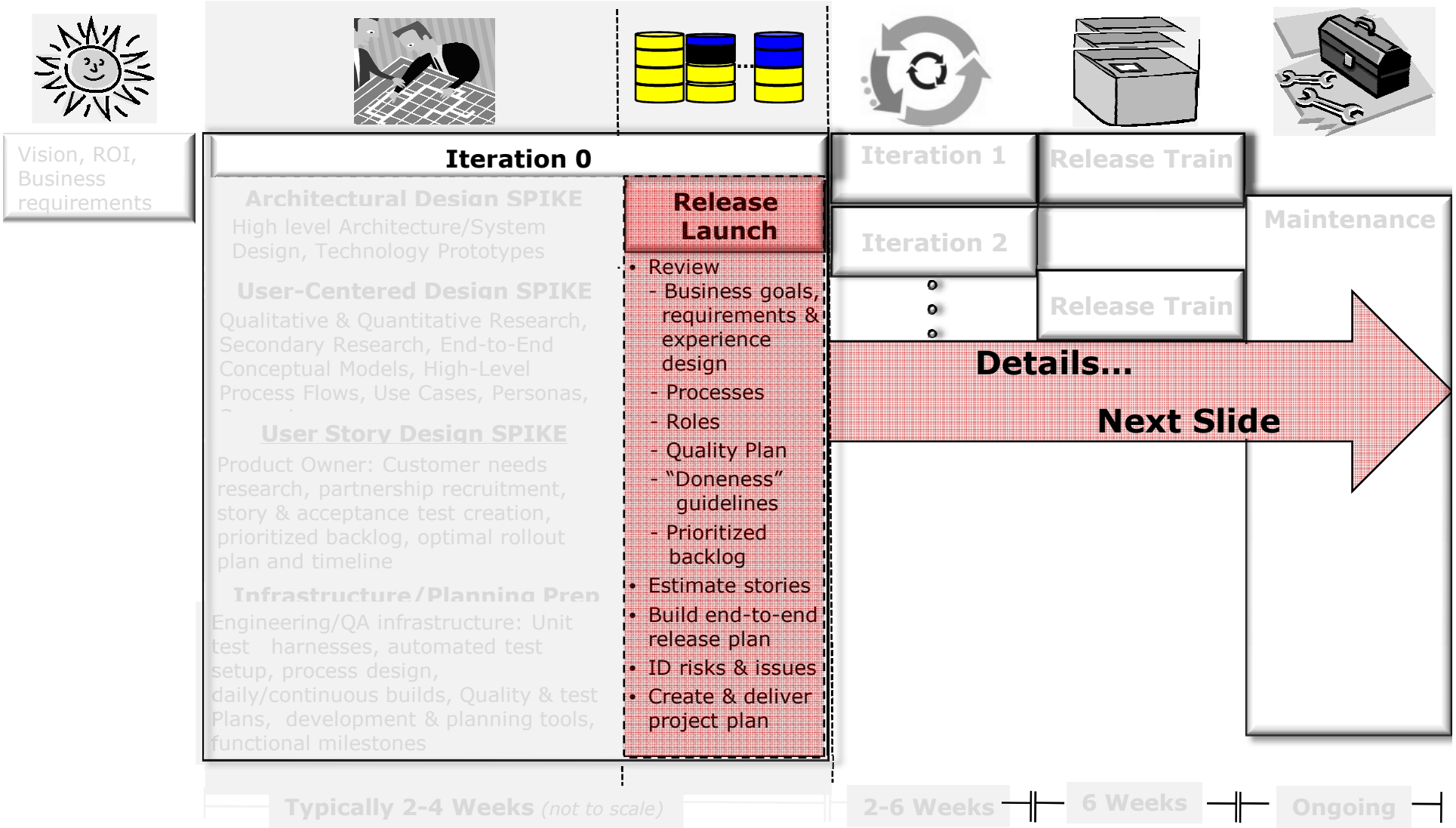
intuit

TSP Agile-Blend Model Detail



* Special thanks to Glynis Hively for her User-Centered Design contributions.

TSP Agile-Blend Model Detail



TSP-Agile Blend Release Launch

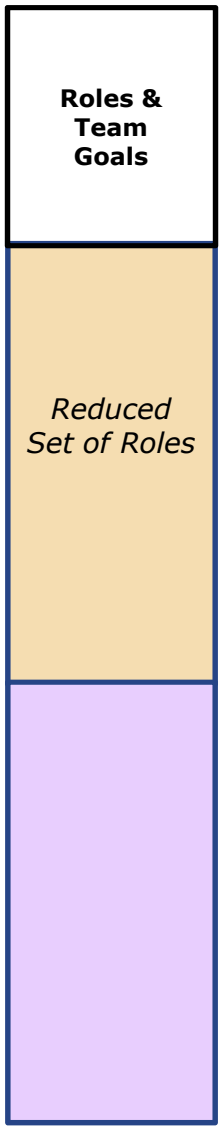
Driven more by agile thinking
Driven more by TSP thinking

Product & Business Goals	Roles & Team Goals	Development Strategy --- Conceptual Design --- Sizing	Top Down & Next-Phase Plans	Bottom-up & Balanced Plans	Quality Plan	Risk Assessment	Briefing & Launch Report	Management Review
	Reduced Set of Roles	*StoryPoint Estimation* *Agile Processes & Practices*	*StoryPoint-Based End-to-End Release Plan without a detailed WBS* *WBS for next cycle only with Tasks, Hours, & Assignments*		*Definition of "Done"* *Less emphasis on defect analysis*			
		Iteration 0 Architecture & High-Level Design			*More rigor with metrics*			

Significant differences from a typical "Waterfall" Project in the TSP Framework

Significant differences from a typical "Agile" Project

Roles



intuit.

Roles

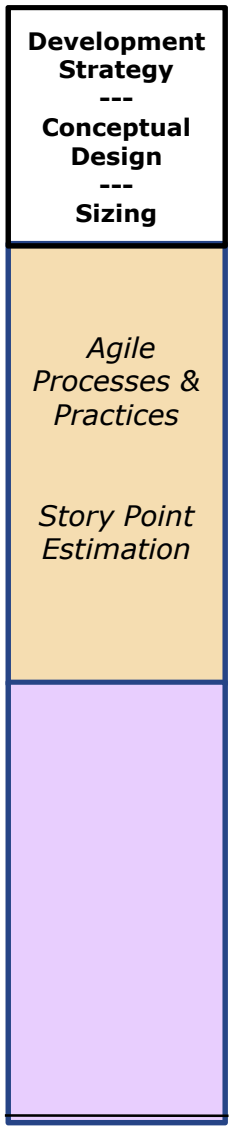
- **ScrumMaster** — Team Leader, Planning Mgr, Process Mgr
  - Makes sure Scrum rules are followed
- **Scrum Team** — Team Member
  - Self-managed, cross-functional team builds the product
- **Product Owner** — Customer Interface Manager
  - Sets release plan, iteration goals & content & acts as a customer proxy

Other TSP roles to consider:

- **Quality Manager**
- **Test Manager**
- **Implementation Manager**
- **Design Manager**
- **Support Manager**

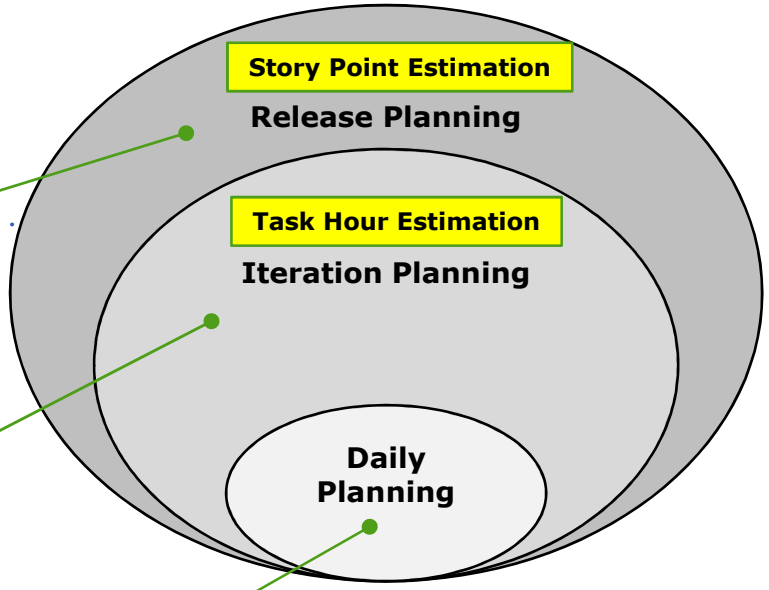
Traditional Scrum roles
TSP roles

Story Point Estimation



Planning Levels

- Release planning
  - End-to-end plan
  - Done at Iteration 0 launch then <u>continuously</u> refined
- Iteration planning
  - During launch and each re-launch
  - Once underway– no changes are allowed unless spawned by the team itself
  - Repeatedly done for the "current launch/re-launch"
- Daily planning
  - During execution of each iteration



User Stories

- Describes something of value to the user of the system:
  - Has a role and a specific outcome
    - *Role, Requirement, Intent*
    - *As a ___, I want to ___, so that I can ___*
  - Is testable
  - Is estimable ("100 days or instead, x in 15 days")
  - Is prioritized (by the user)

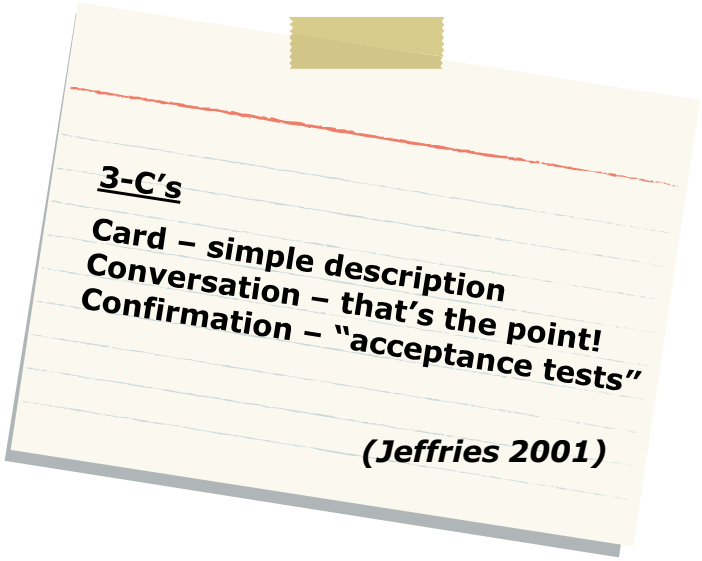
Optional if obvious

- Example:

  As a "Preferred Gold" member,

  I want to convert my options to points,

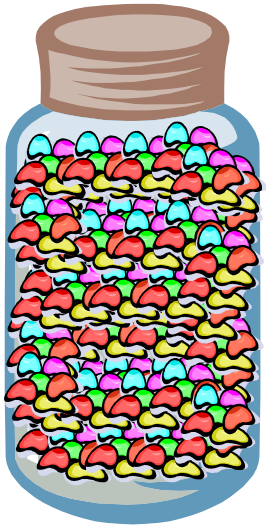
  so that I can stay in international hotels.



My Favorite Reference: Mike Cohn, "User Stories Applied"

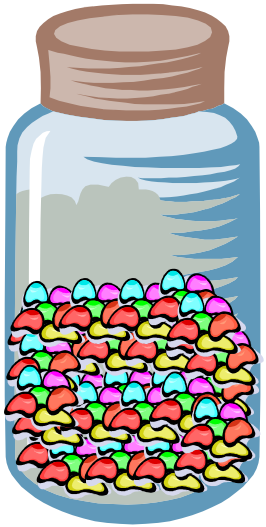
Story Points

- An unit of relative measure for expressing the over all size of a user story
- Influenced by difficulty, risk, complexity, et al

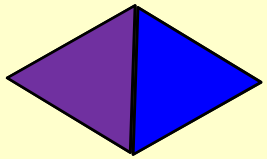


Which is easier to answer?

1. **How many jelly beans are there in Jar 1? In jar 2?**
2. **How many jelly beans are there in Jar 1 relative to Jar 2?**



TSP
Earned Value



Agile
Story Point Burndown

The Same
But Different

intuit

Story Point Estimation with Planning Poker

Team discusses User Story

Each person chooses their estimate

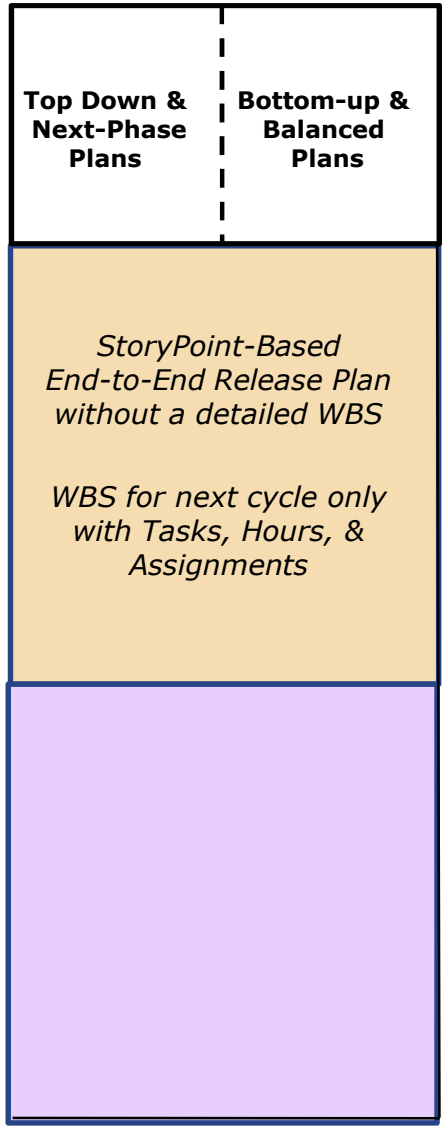
Simultaneously, each person shows estimate

People with high & low estimates explain their reasoning

Until the numbers are close

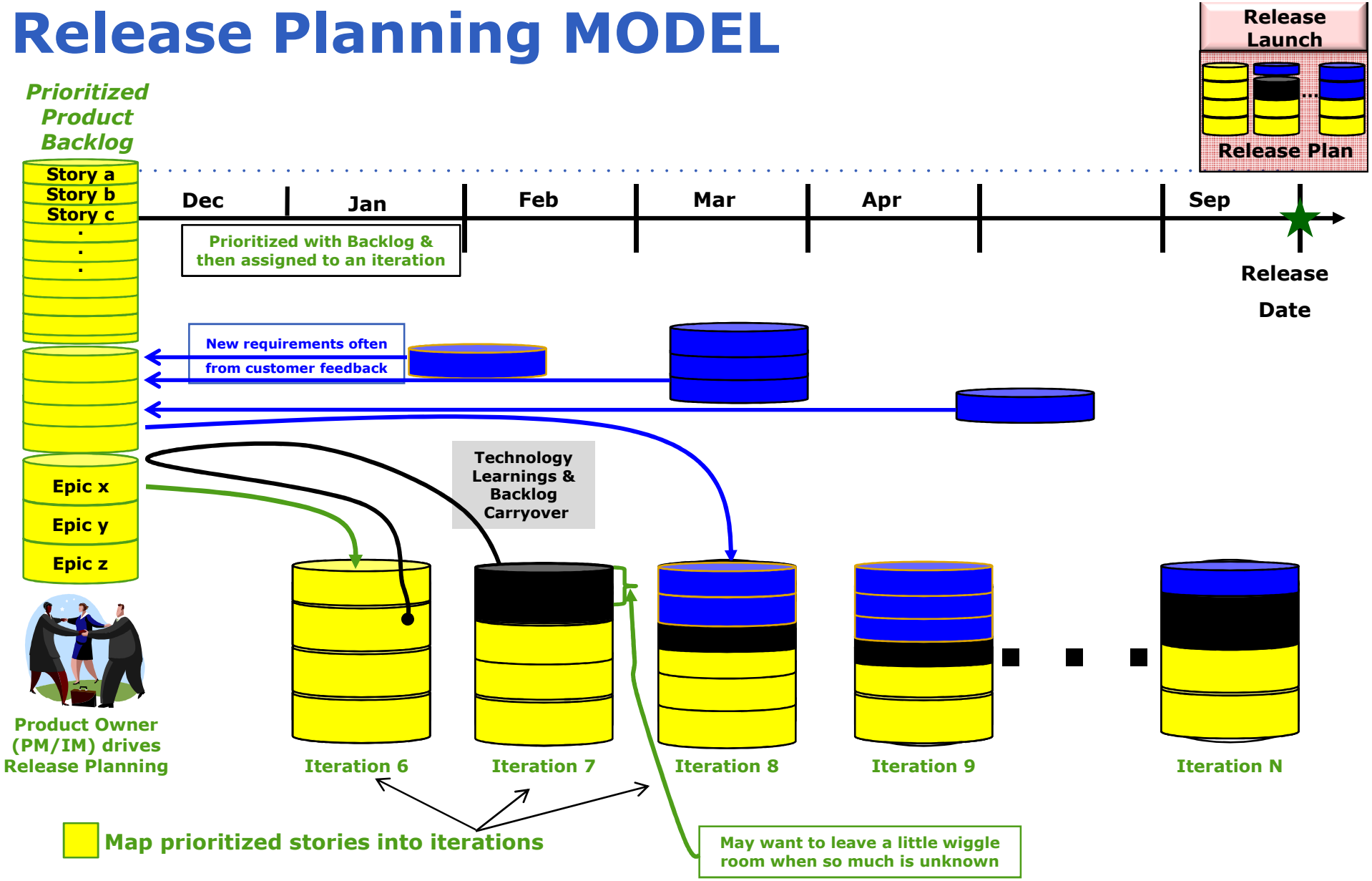
intuit

Release Plan



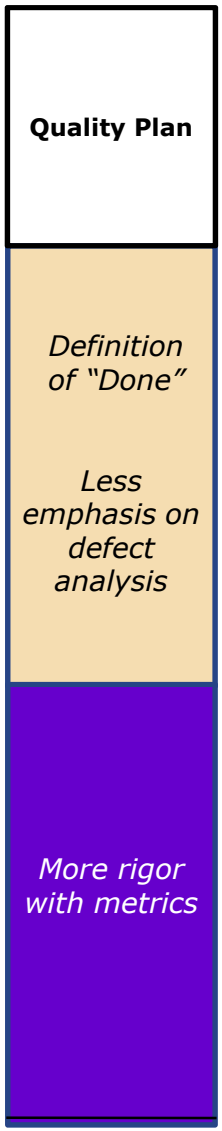
intuit

Release Planning MODEL



intuit

"Doneness"



intuit

Definition of "Done" – Metrics Only*

Schedule Metrics

- **Iteration Burndown –** Task Hours
- **Release Burndown –** Earned Value (EV) or Story Points
- **Task Hours –** Planned (P)/Actual (A) Hours
- **Story Points or EV –** P/A

Scope Metrics

- **Scope** – EV Growth/Time or Story Point Count

Productivity[1] Metrics

- **Productivity –** LOC/Hour or Story Points/Iteration (Velocity) – Task Hours/Week

Quality Mindset

- Working, ready-to-ship software that meets or exceeds customer expectations
- Quality is everyone's job & is continuously built-in
- Two customers must be served:
  1. End-user
  2. Funding organization

Quality Metrics

Reviews

- **Reviews: Time, Defects Found**[2]
- **Design : Review Time**
- **Code : Review Time**
- **Test : Test Review Time**
- **Design : Code Time**

Test

- **Test Code Coverage %**[3]
- **Feature/Story Coverage %**
- **Test Execution – Results**
- **Test Automation – %**

Bugs

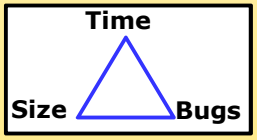
- **#/KLOC**
- **# Incoming, Open, Closed**
- **# Open by Impact & Severity**
- **# Post-release defects**

Optional[4]

- Defect Type
- Defect Phase Insertion
- Defect Phase Detection

"Still" Derivable

- Cost of Quality
- Review Rates
- Yield
- Defect Cost
- Defect Predict



[1] *For planning only – never used in performance reviews!*

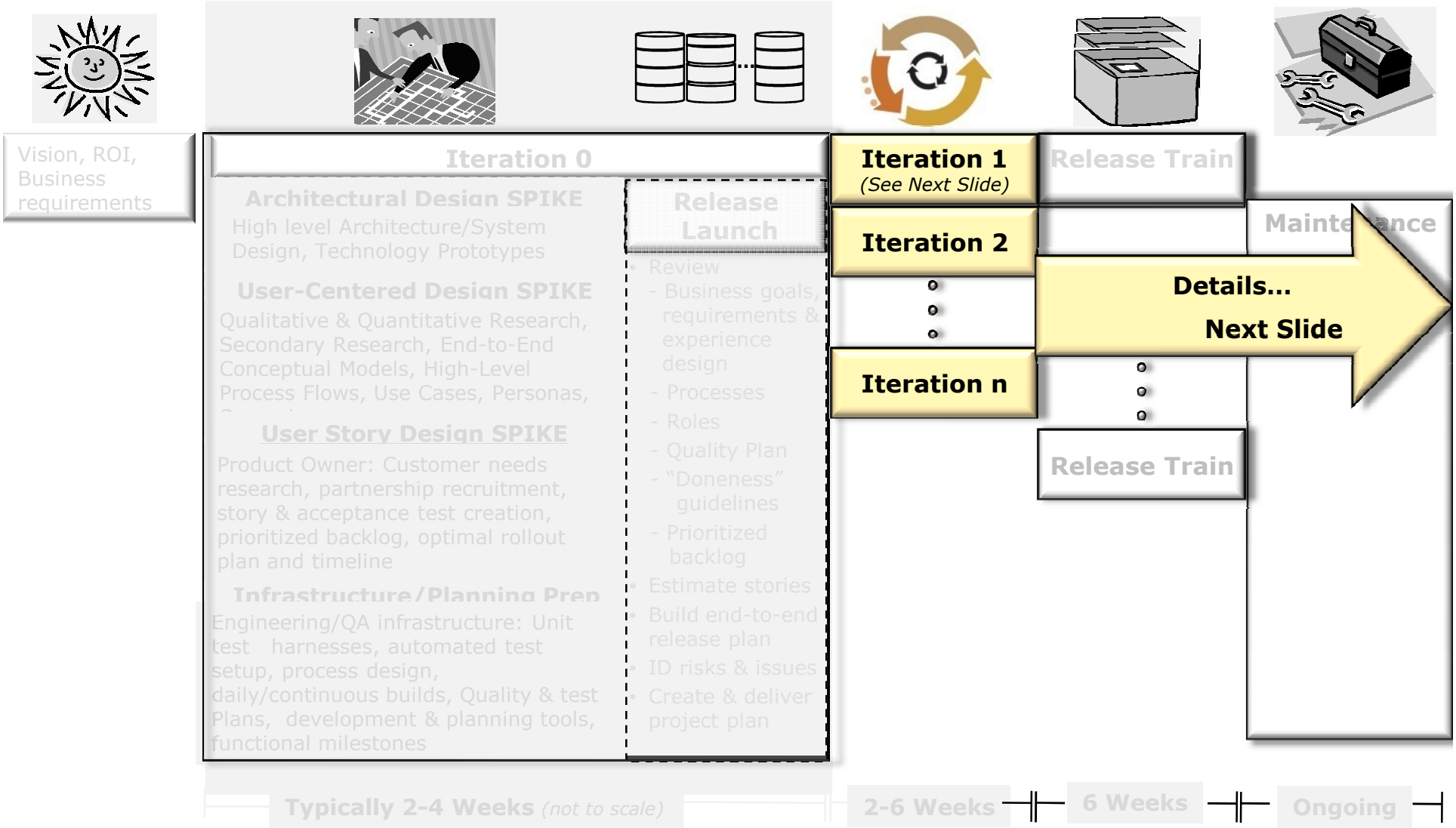
[2] *- Reviews may include collaborative design or pair programming time which are common agile practices.*
- Planned (P) measures for Quality are desirable but optional.
- Significant (non-cosmetic) defects only

[3] *Test Code Coverage includes unit & black box testing*

[4] *A valuable empirical method to help determine the most important areas for improvement is lost*

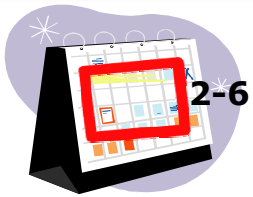
**There is other criteria besides these type of metrics that go into defining "done" for agile projects*

TSP Agile-Blend Model Detail



Iteration Process

Launch



Execute

Review

- Define iteration goal
- Identify desired stories
- Create WBS for stories
- Define "Doneness"
- Commit

- Do tasks
- Be part of Daily Standup

- Present results
- Demo code
- Conduct retrospective
- Deliver working code, tests, & results
- Update end-to-end Release Plan

SUMMARY

- Perceptions about TSP & Agile are generally skewed --- training & education help
- TSP is a terrific framework that most Agile principles and practices fit into very nicely
- Blending the two can give you the best of both worlds enabling you to:

Create high-quality products
in a predictable and repeatable fashion

including projects with
rapidly changing or
unknown customer or
technology requirements

Contact Information

Alan Padula
alan_padula@intuit.com