regoUniversity

Introduction to Agile Capitalization

Your Guides:

Rob Greca and Tad Beaty

Agenda

- Agile Capitalization in Software Development Overview of Agile
- Agile Capitalization Primer
- Accounting Practices and Agile
- The Problem with Tracking Programmer Hours
- Transitioning from Waterfall to Agile Capitalization
- Questions

Introductions

- Take 5 Minutes

- Turn to a Person Near You

- Introduce Yourself

- Business Cards

Part I:
Agile Capitalization in Software Development

Understanding Software as a Capital Investment

Defining Capitalization

Capitalization means spreading investment costs over a long-term asset's lifetime. Capitalized expenses become part of a company's listed assets and are used in financial reporting and tax filings.

Introduction: Software Development as an Investment

- Investing in software development is a long-term commitment.
- Up-front costs on engineer salaries lead to future profit.
- Investing wisely converts cash into software, raising company value.

Distinguishing Long-Term Investments in Software

- Not all software development projects are long-term investments.
- Criteria depend on whether the software remains an asset.
- Mistakes in classification can result in overpaying taxes and understating value.
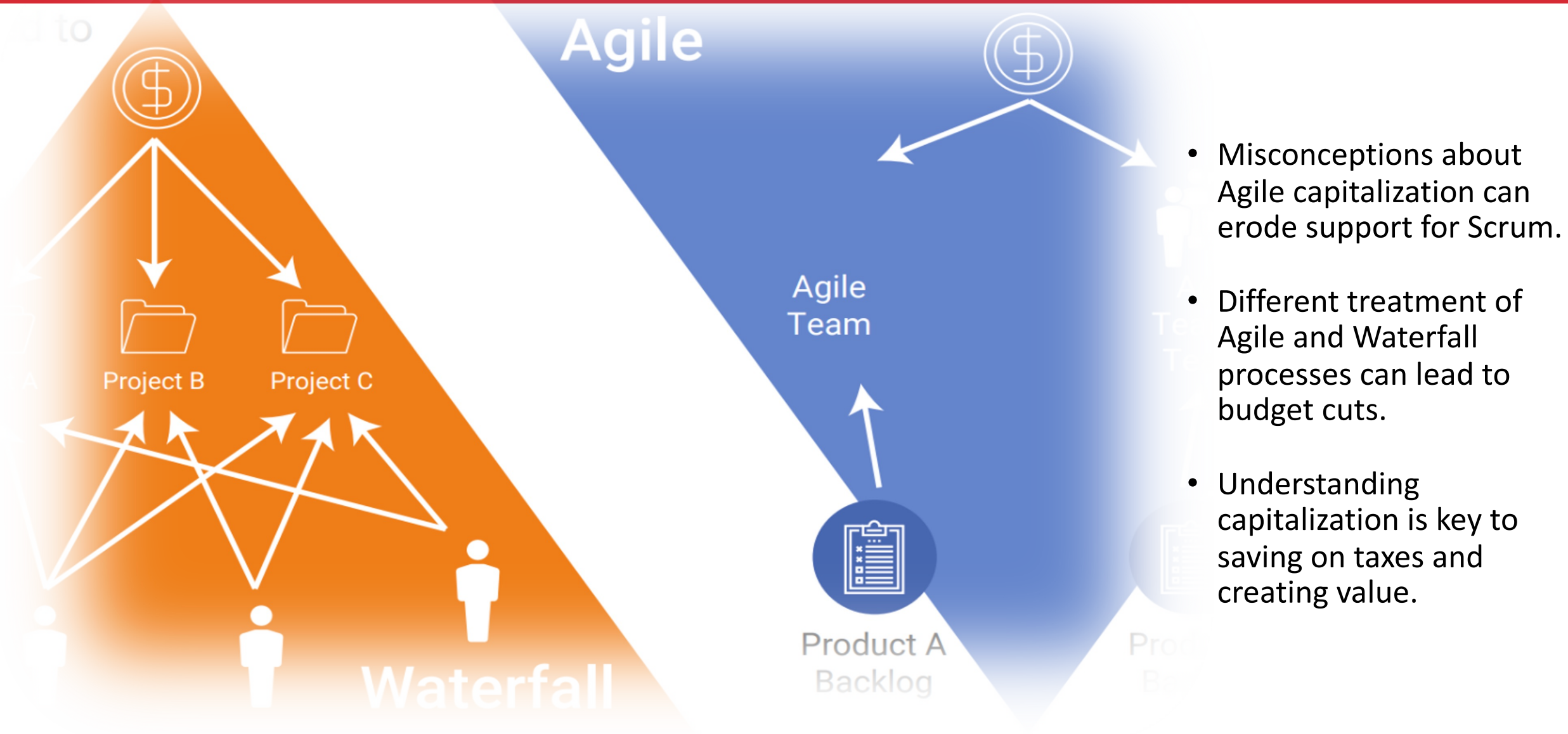
Potential Pitfalls & Conclusions

- Incorrectly treating software investment can hide its true value.

- Consequences include depressed stock price and reduced borrowing power.

- Recognizing software as a capital investment aligns with its true value and potential.

Part 2: Agile Capitalization Primer

Why Agilists Should Understand Capitalization

The Why…

- Misconceptions about Agile capitalization can erode support for Scrum.

- Different treatment of Agile and Waterfall processes can lead to budget cuts.

- Understanding capitalization is key to saving on taxes and creating value.

The Mistreatment of Agile Development in Finance

- Some companies treat Agile as an operational expense and Waterfall as capital.

- Misunderstandings have led to millions lost in improper taxation.

- Poor capitalization rules make Agile companies appear poorly managed.

The Importance of Properly Capitalizing Agile Development

- Proper capitalization aligns finances responsibly with shareholders and regulators.

- Agile methods can track labor costs more accurately than Waterfall processes.

- Misapplied Waterfall language can lead to misclassification.
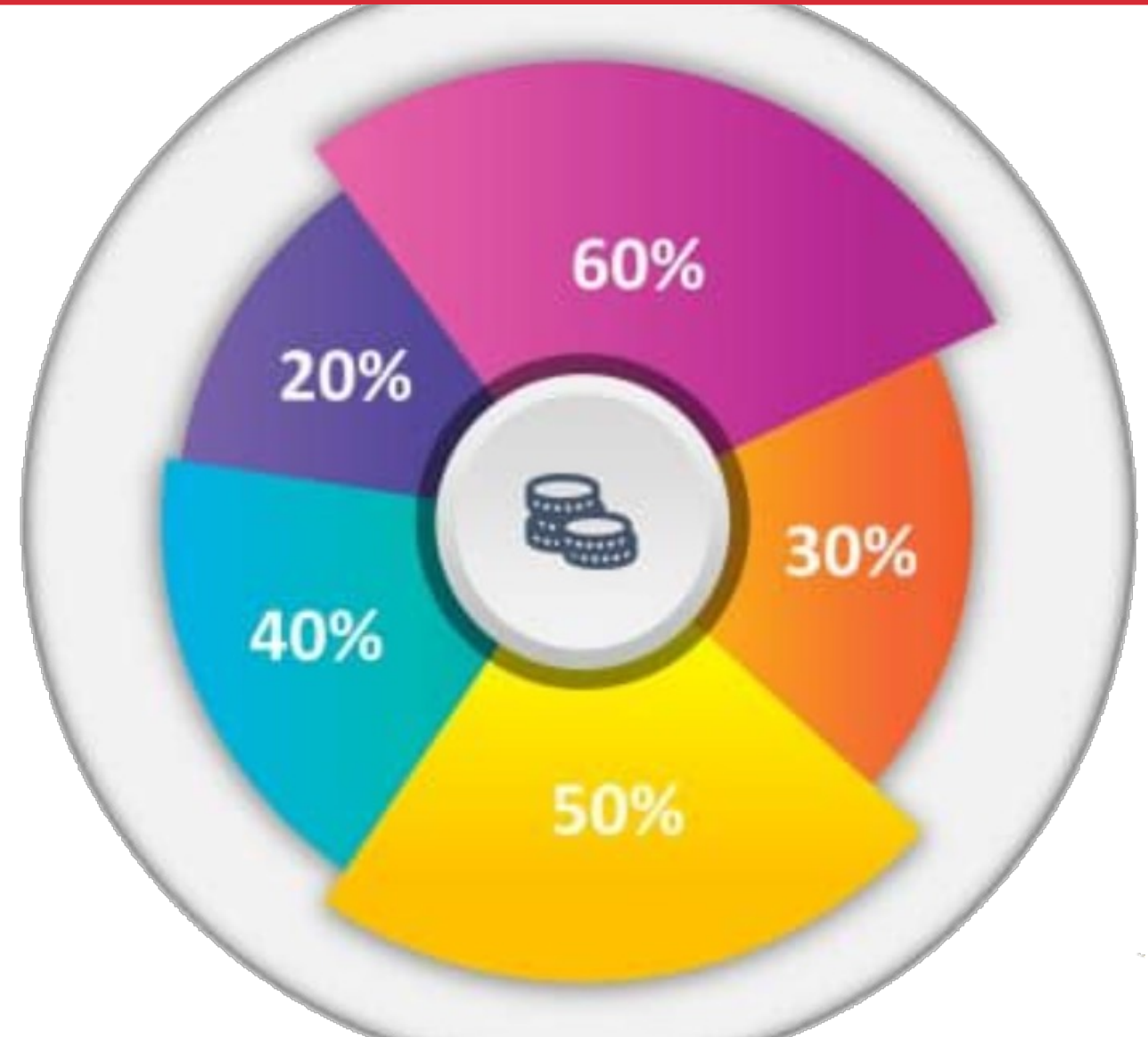
Risks of Misunderstanding Agile Capitalization

Incorrect treatment can lead to excessive tax liability and staff cuts.

Risk of audit exception findings and subsequent re-reporting of earnings.

Departments potentially expecting headcount to be halved.

The Power of Scrum Masters Understanding Capitalization

Scrum Masters can classify work accurately, defend classifications, and align behavior with goals.

Good agile practices enable more verifiable capitalization, reducing tax and funding additional engineers.

Scrum Masters may become experts in software capitalization, depreciation, and impairment.

Agile and Finance, Better Together

Agile capitalization is not merely a financial concern but a strategic opportunity.

Educating both finance and development teams ensures alignment and promotes growth.

Embracing Agile capitalization principles can lead to more responsible reporting and substantial savings.

Part 3: Accounting Practices and Agile

Bridging the Gap between Accounting and Development

The Regulatory Landscape

US Financial Accounting Standards Board (FASB):

Under U.S. GAAP (Generally Accepted Accounting Principles), governed by the FASB, the capitalization of software development costs is guided by the ASC 350-40, which stipulates how to handle internal-use software, which would include Agile projects. The guideline divides the software development process into three stages:

- Preliminary Project Stage: Costs incurred during this stage should be expensed as they are incurred.

- *Application Development Stage: Costs incurred during this stage can be capitalized. This includes coding, hardware installation, and software configuration. <u>In Agile, this typically encompasses the build and code work done during Sprints</u>.*

- Post-Implementation/Operational Stage: Costs incurred during this stage should be expensed as they are incurred.

International Accounting Standards Board (IASB):

Under International Financial Reporting Standards (IFRS), governed by the IASB, the capitalization of development costs is guided by the IAS 38, which details the accounting treatment for intangible assets, including software development costs. The guideline generally allows for capitalization of costs if, and only if:

- It is technically feasible to complete the intangible asset.

- The entity intends to complete the intangible asset and use or sell it.

- It is possible to demonstrate how the intangible asset will generate probable future economic benefits.

- The entity has sufficient resources to complete the development and to use or sell the intangible asset.

- The entity is able to reliably measure the expenditure attributable to the intangible asset during its development.

Agile Principles Align with Compliance and Finance Goals

Regular Reassessment of Assets and Expenses: Agile principles encourage iterative planning and frequent reassessment, which aligns well with the accounting goal of continuously evaluating the nature of expenditures to correctly categorize them as capital or operational expenses.

Dynamic Allocation of Resources: Agile enables dynamic allocation of resources which can help in making informed decisions about capitalizing or expensing particular costs, and quickly adapting to changing regulatory or business requirements.

Integration of Accounting Perspective: Cross-functional teams can integrate accounting expertise from the beginning of a project, facilitating correct categorization of costs (as capital or expense) based on real-time input and collaboration, thus enhancing the accuracy of financial statements.
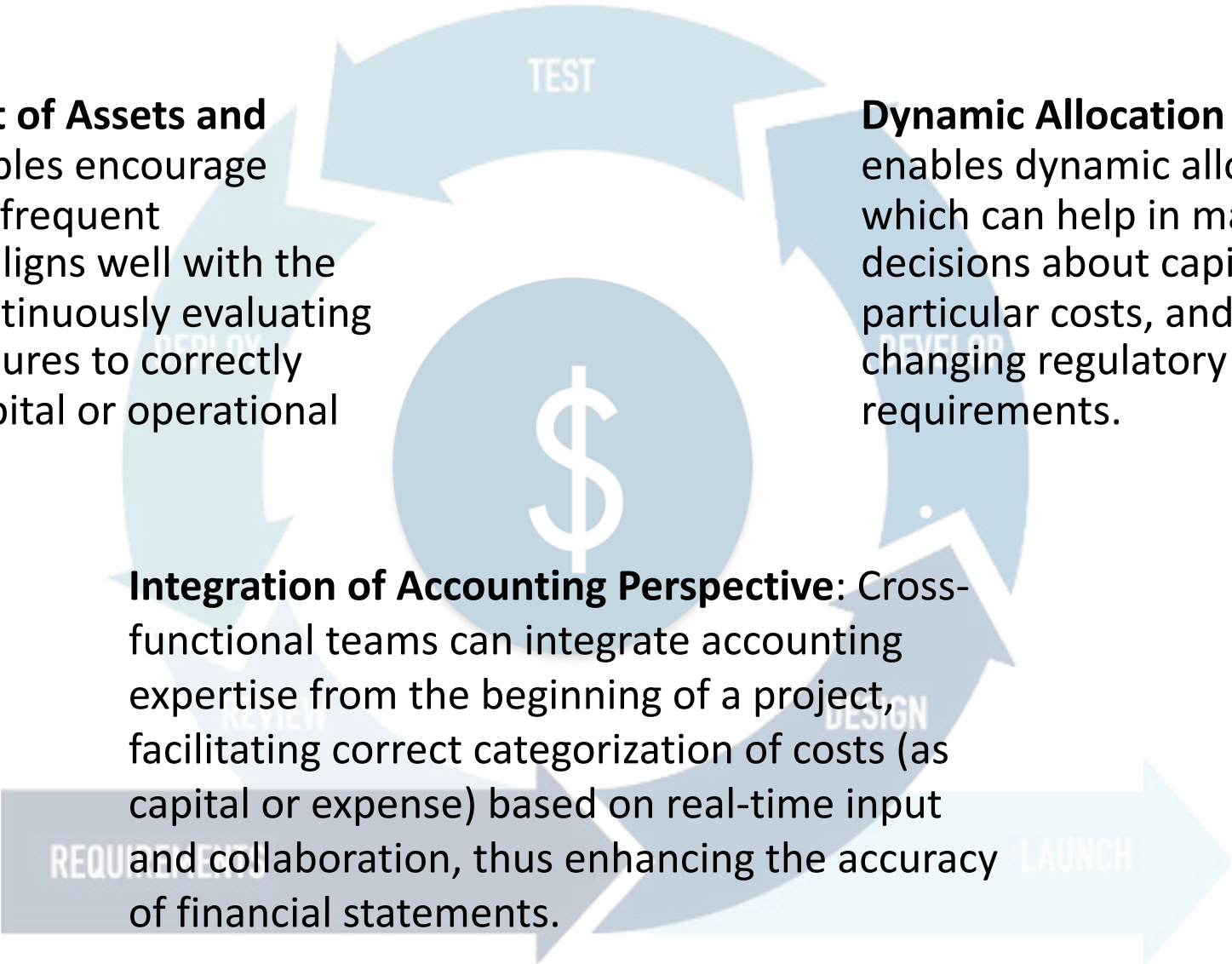
Establishing the Capitalization Process

Establish the Bright Line: Determine when your company starts capitalizing work. This is after the "Preliminary Project Phase," once committed to funding, and probable completion.

Start of Capitalization: Often begins with the first Sprint after completing initial market exploration and architectural design.

Feasibility Spike Sprint: Costs related to initial research are considered operational expenses.

Inclusion: Engineers, testers, designers, product and project management, and Scrum Masters can all be capitalized.

Mixed-Mode Projects and Activities

Criteria for Mixed-Mode Projects: Identify whether the entire project or only part of it should be capitalized.

Common Expensed Activities: Include fixing bugs in released products, localization for international releases, data conversion, training, operations beyond deployment, SOX or security reviews, refactoring, and modifications for individual customers.

Consultation with Finance Department: Emphasize the importance of working with the finance department and technical accounting advisors to determine capitalization or expensing for specific items.

Apportioning Labor to Mixed-Mode Projects

Defining Mixed-Mode Projects: Establish the process for dividing labor into capital or operational expenses using Scrum practices, including proportional allocation of estimation points or "story points."

1. Identify and Segregate Agile and Traditional Components including Labor and Operational Costs

2. Sum the total Agile and Traditional Costs

3. Identify and sum of the total story points completed in the Agile Component of the project (generally found in the Agile development management tool)

4. Integrate the cost per point data with the traditional data to derive insights for the mixed-mode project.

$$\text{Cost Per Point (Agile Component)} = \frac{\text{Total Agile Component Costs}}{\text{Total Story Points}}$$

Utilize Product Backlog Items to Document Capitalization Justifications

Feature Example:

Title: Enhancement of Search Functionality to Include Voice Recognition

Acceptance Criteria:

1. The voice recognition system can accurately recognize and process voice commands in various accents and languages with at least a 95% success rate.

2. The system should be able to handle multiple simultaneous voice command requests without any significant delays.

3. The voice recognition feature should be compatible with both the mobile app and web platform.

4. Compliance with data protection and privacy regulations to ensure the security of user data.

Capitalization Justification:

1. **Future Economic Benefits**: This feature is expected to increase user engagement and retention, potentially driving increased sales over the long term.

2. **Enhancement of Functionality**: The development of this feature involves significant enhancements to the existing platform, adding a new functionality that is expected to deliver benefits over several years.

3. **Amortization Over Expected Useful Life**: The costs incurred in the development of this feature can be amortized over its expected useful life, aligning with capitalization criteria.

Part 4: The Problem with Tracking Programmer Hours

rego University
SAN DIEGO · 2023

Why Tracking Programmer Hours is a Mistake

- **Disruptive Nature**: Hourly time tracking disrupts Agile behavior and the creative process of software development.

- **Inaccuracy and Unverifiability**: Late tracking (such as at the end of the week) leads to loss of detailed information and results in inaccurate reporting.

- **Auditors' Perspective**: Auditors recognize the challenges of hourly tracking.

A Better Approach - Proportional Allocation by Story Points

Estimating Effort vs Actual Time: A high correlation exists between estimated effort and actual work time in Scrum, highlighting its forecasting accuracy.

Supported by Auditors: Educated auditors become enthusiastic supporters of proportional allocation as it offers honest transparency and well-documented reporting.

The Result: Tracking Product Backlog Items (PBIs), estimation points, and completion dates results in a clear understanding of the work, supported by team members and executives alike, creating an auditor's dream scenario.

# Part 5:Transitioning from Waterfall to Agile Capitalization

Changing the Financial Reporting Landscape

Leverage Your Transition: The transition from Waterfall to Scrum as an ideal time to re-evaluate financial reporting practices.

Eliminating Hour Tracking: Agile practices allow for a system that doesn't need to track actual hours, and this benefits not only developers but auditors and financial staff.

Increased Capitalization: This more accurate and responsible approach might lead to an increase in capitalized work, which is an expectation within Agile practices.

Potential Red Flags: Note that dramatic changes can be seen as red flags by finance departments and auditors, requiring careful communication.
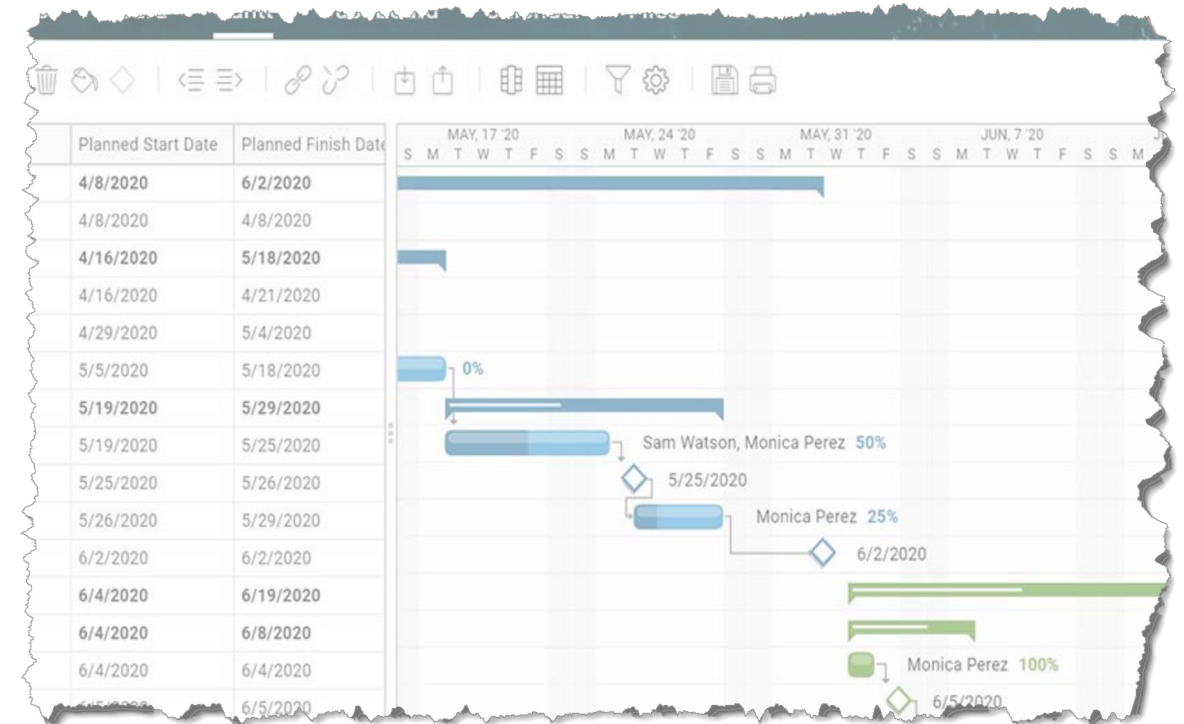
Collaboration and Transparency with Finance

Open Communication is Key:

- The transition to Agile capitalization is a mindset shift.

- Be frank about any expected changes in capitalization rate in conversations with Finance and Audit.

- Communicated the link between these changes and the transition to Agile.

Questions?

Surveys

Please take a few moments to fill out the class survey.
Your feedback is extremely important for future events.

Thank You For Attending Rego University

Instructions for PMI credits

- Access your account at pmi.org
- Click on **Certifications**
- Click on **Maintain My Certification**
- Click on **Visit CCR's** button under the **Report PDU's**
- Click on **Report PDU's**
- Click on **Course or Training**
- Class Provider = **Rego Consulting**
- Class Name = **regoUniversity**
- Course **Description**
- Date Started = **Today's Date**
- Date Completed = **Today's Date**
- Hours Completed = **1 PDU per hour of class time**
- Training classes = **Technical**
- Click on **I agree** and **Submit**

Let us know how we can improve!
Don't forget to fill out the class survey.

Phone
888.813.0444

Email
info@regoconsulting.com

Website
www.regouniversity.com