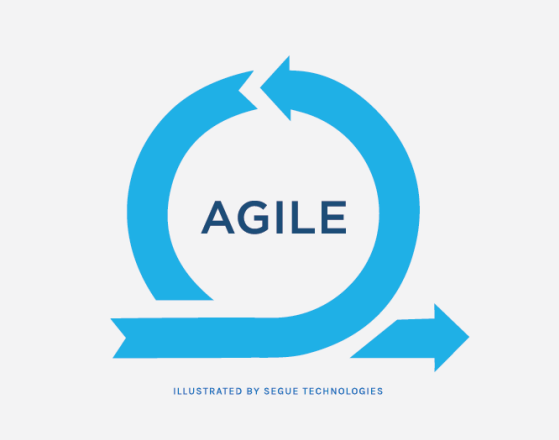
The Fact that your Project is Agile Is Not Necessarily a Cost Driver

Arlene Minkiewicz
Chief Scientist, PRICE® Research

Estimate with Confidence™

Agenda

- Introduction
- Agile Overview
- What the Fact of Agile is Not Enough to Predict Cost
- Methodology and Rules of Thumb for Estimating Agile Projects
- Useful Agile Metrics
- Conclusions

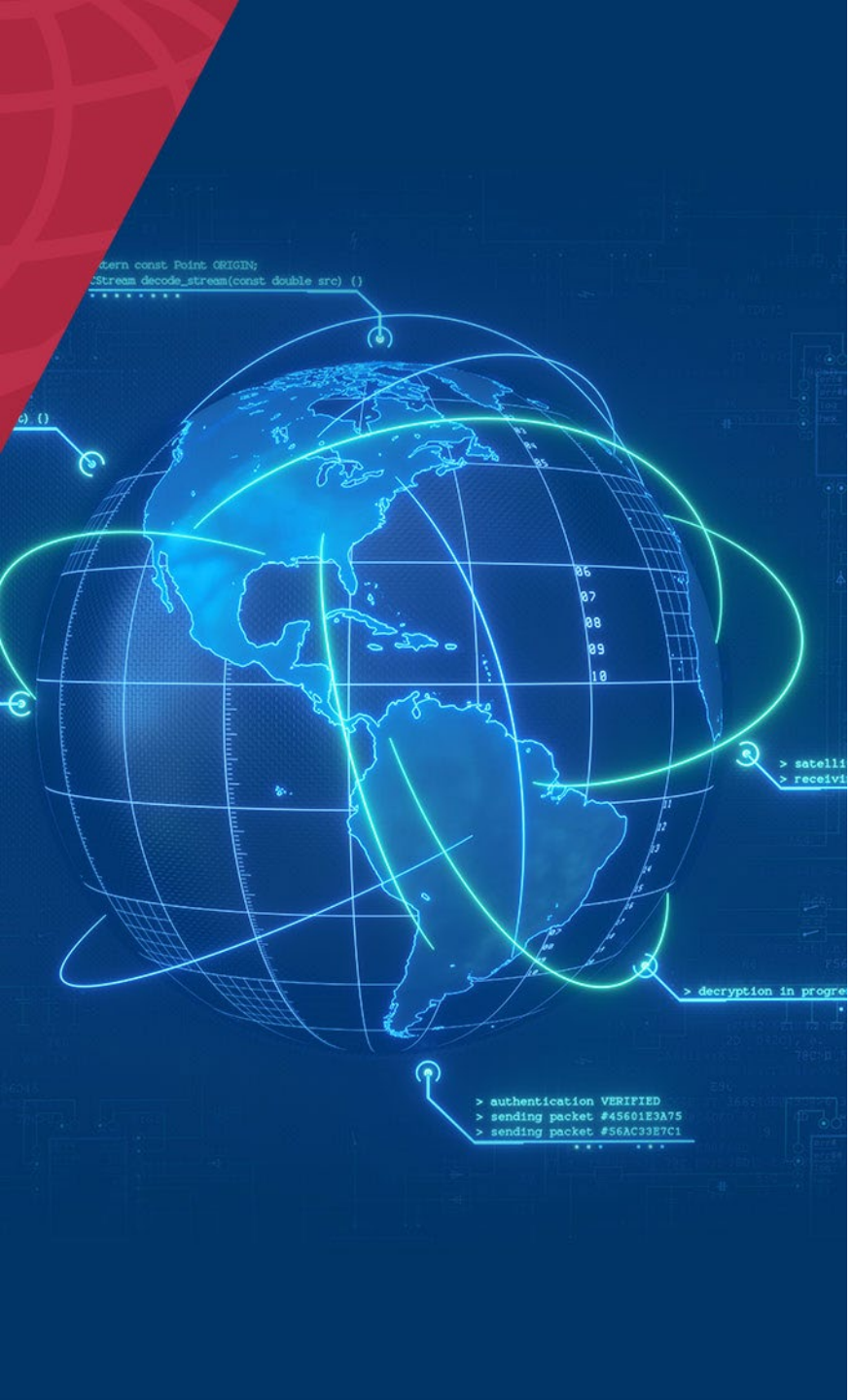


Introduction

- Agile development practices have enabled organizations to deliver quality software that optimizes customer satisfaction
- Agile processes rely on highly skilled developers communicating with clients and each other to optimize value delivered
- This requires a mind shift during project planning
  - Development teams
  - Consumers
- From a purely agile perspective estimation doesn't make sense
- From a business/program perspective estimation is still important

Introduction

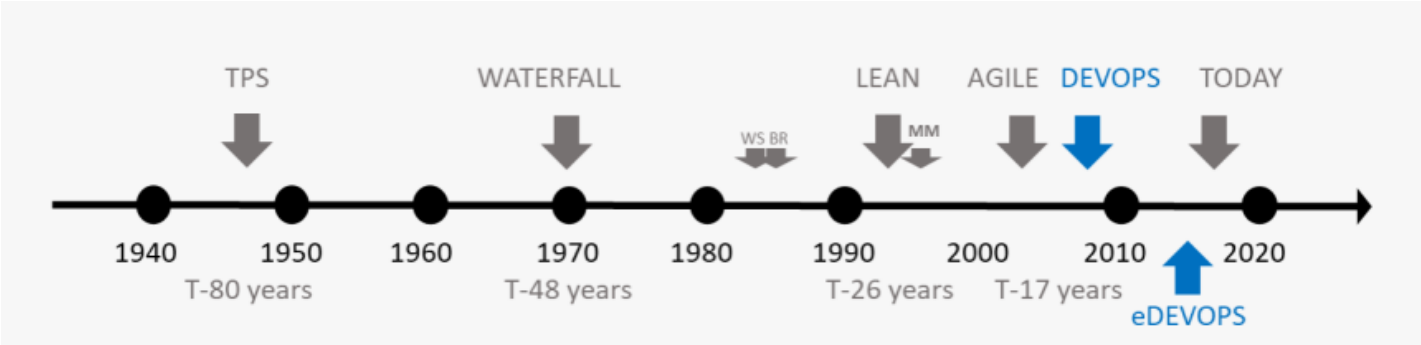
- Agile is a philosophy, not a specific process
  - All agile teams apply this philosophy but not in the same ways
- There are many different frameworks for agile implement, each of which potentially apply:
  - Different Processes
  - Different Practices
- The fact that a project is agile may suggest a different approach to estimation
- Knowing a project is agile is not enough to inform an estimate without conversations between
  - Estimating Team
  - Agile Development Team



Overview

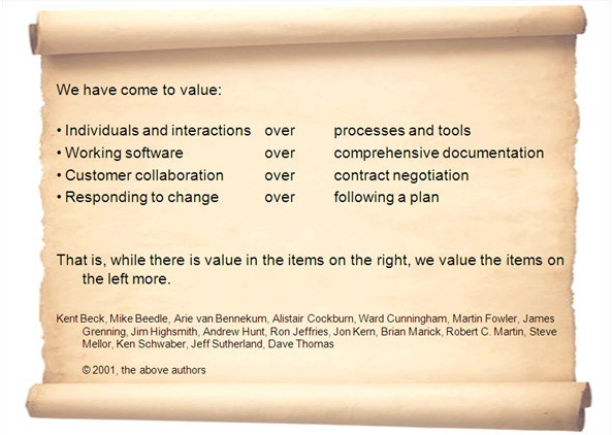
Agile Introduction – History

- At inception, software development was not treated as an engineering discipline but rather a combination of:
  - Art
  - Science
  - With a sprinkle of black magic
- As technology improved, complexity improve:
  - Ensuing failure to step back and take a breath
- Many software projects spiraled out of control



Agile Manifesto

- We are discovering better ways of developing software by doing it and helping others do it
  - Individuals and interactions over processes and tools
  - Working software over comprehensive documentation
  - Customer collaboration over contract negotiation
  - Responding to change over following a plan
- All agile projects adhere to this manifesto
- All agile projects share a common set of principles
- Each agile project uses a unique set of agile practices to implement these principles
- Successful estimation for an agile project is like software estimation for any project – you need to understand the project properties and the practices employed



We have come to value:

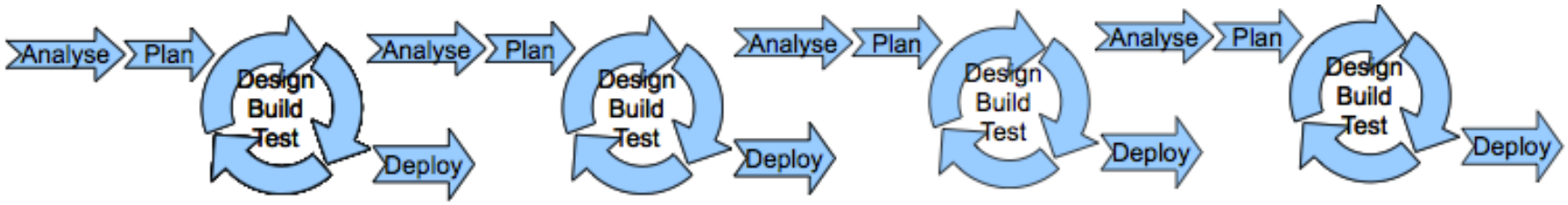
• Individuals and interactions	over	processes and tools
• Working software	over	comprehensive documentation
• Customer collaboration	over	contract negotiation
• Responding to change	over	following a plan

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

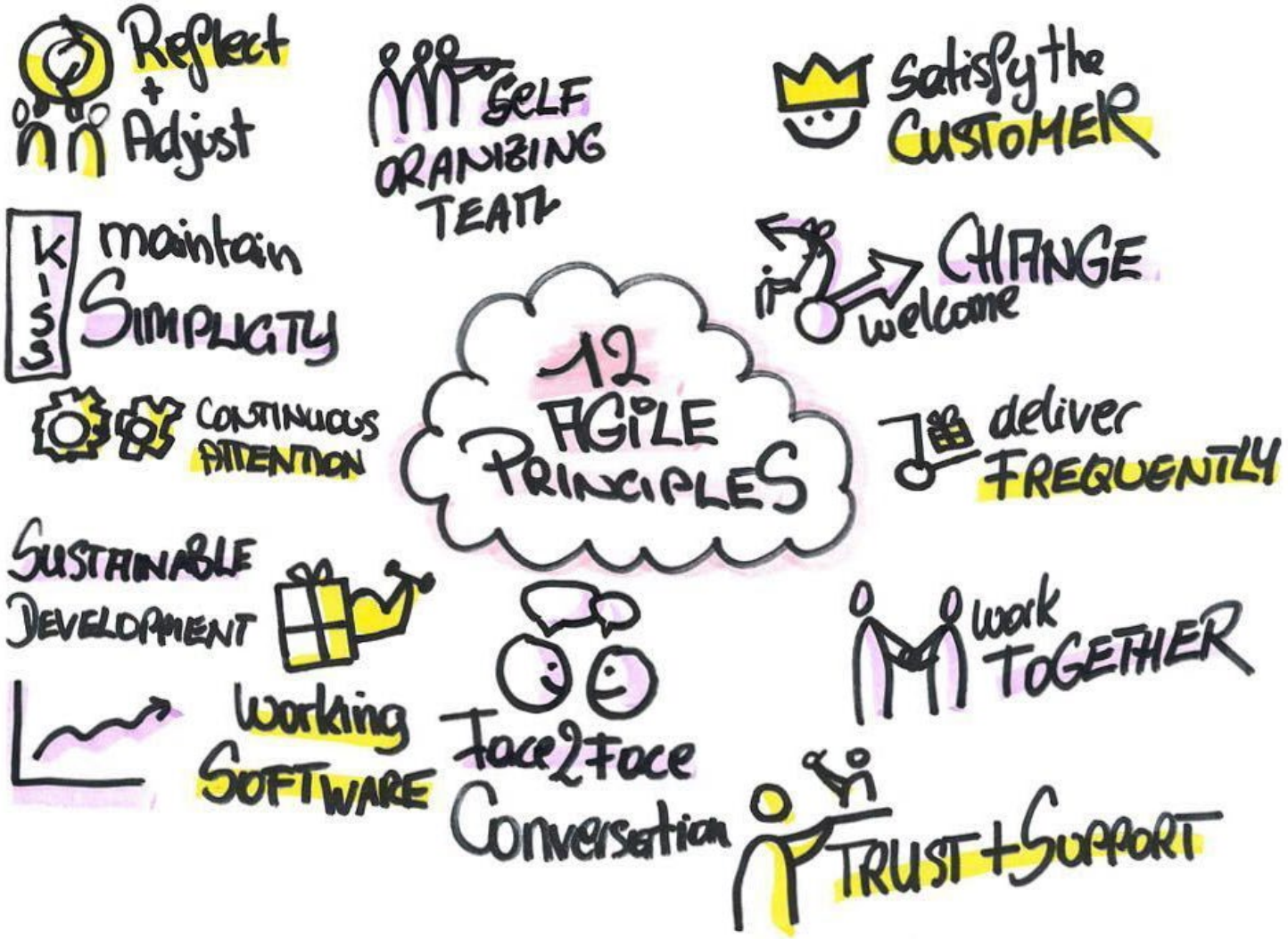
Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas

Agile Software Development

- Usable chunks of software are developed in short periods of time (sprints, iterations, etc.)
- Requirements are translated into user stories and become the project backlog
- User stories deliver business value and are small enough to complete in an iteration
- Customer works with team and reviews software regularly
- Each iteration focuses on the user stories that are currently the highest priority of the customer

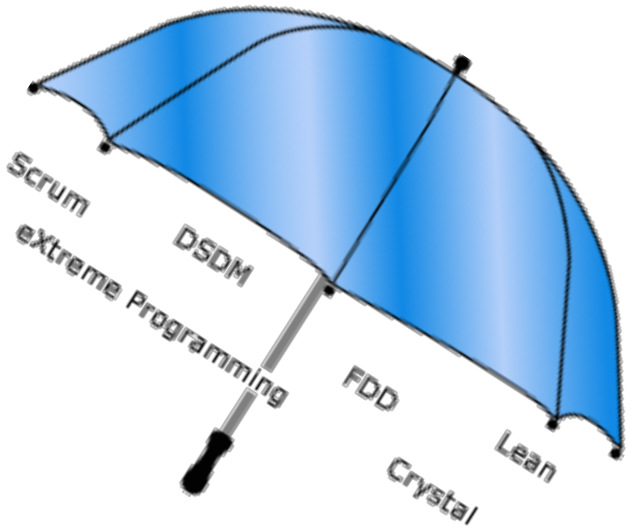


12 Guiding Principles for Agile Development



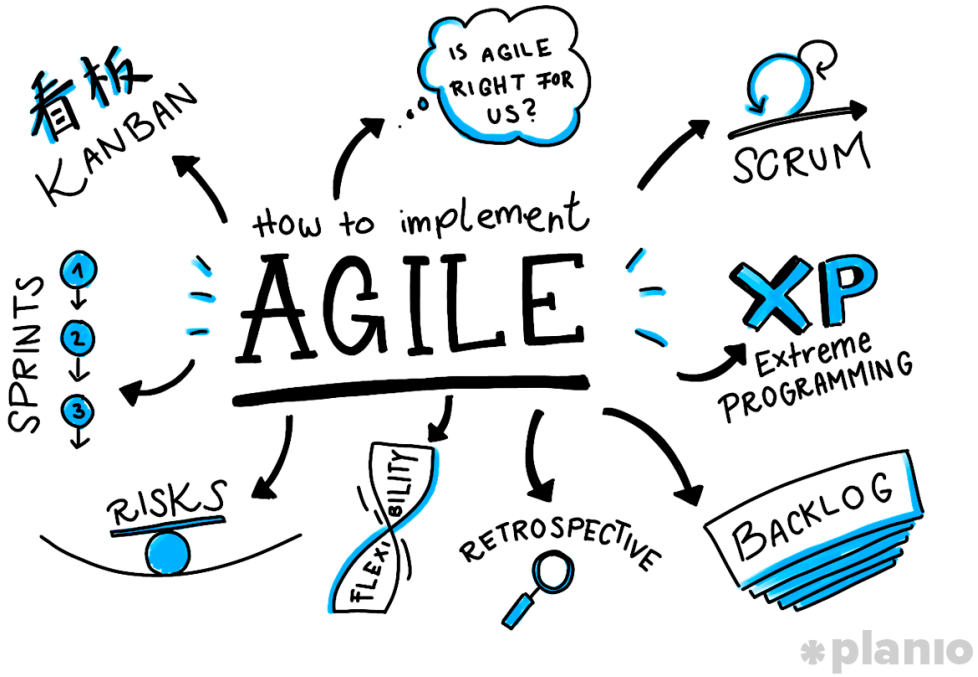
Common Agile Practices

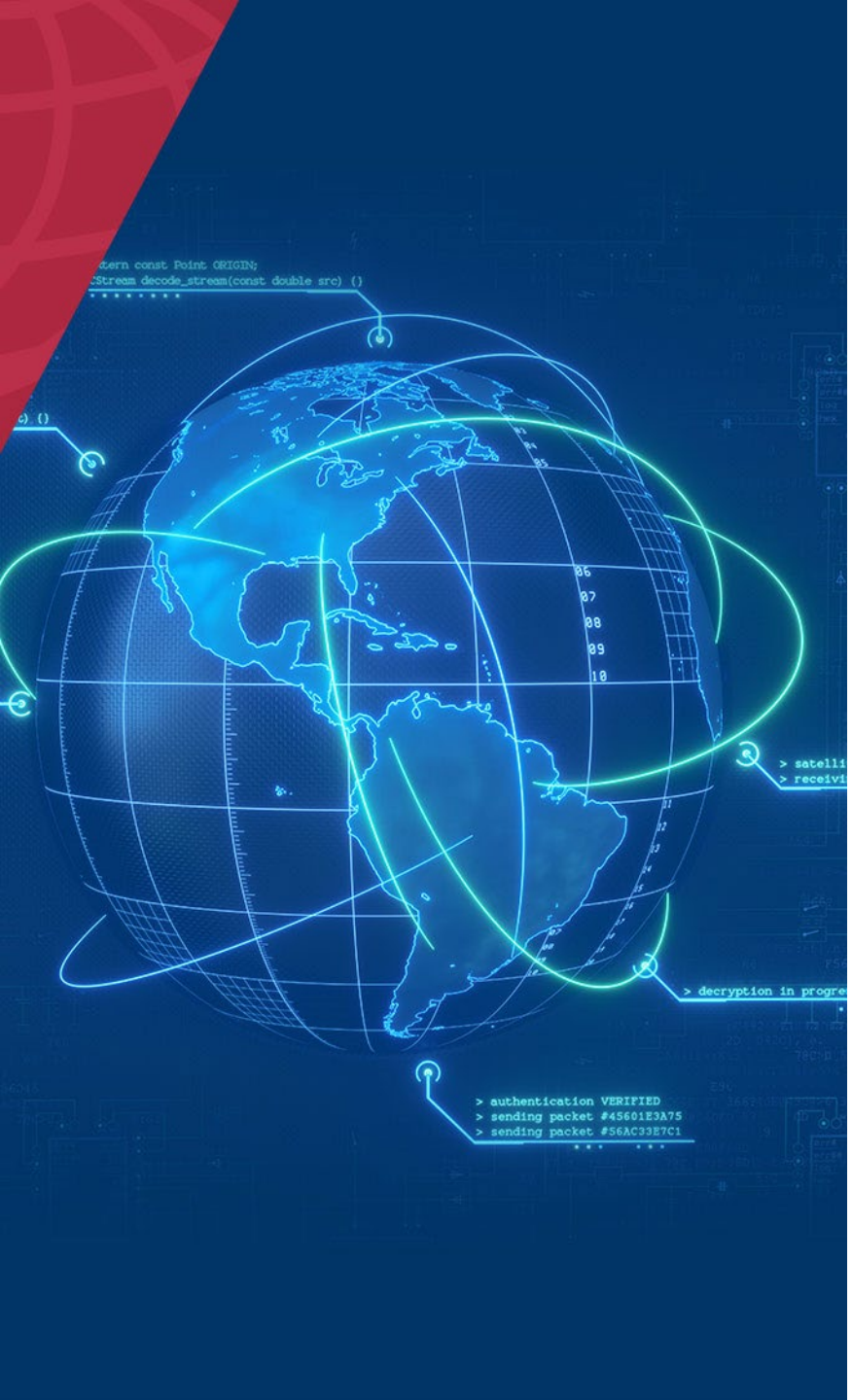
- Pair programming
- Continuous integration with automated testing
- Test driven development
- Daily stand up meetings
- Co-located teams
- Code refactoring
- Small releases
- Customer on team
- Simple design



Common agile frameworks

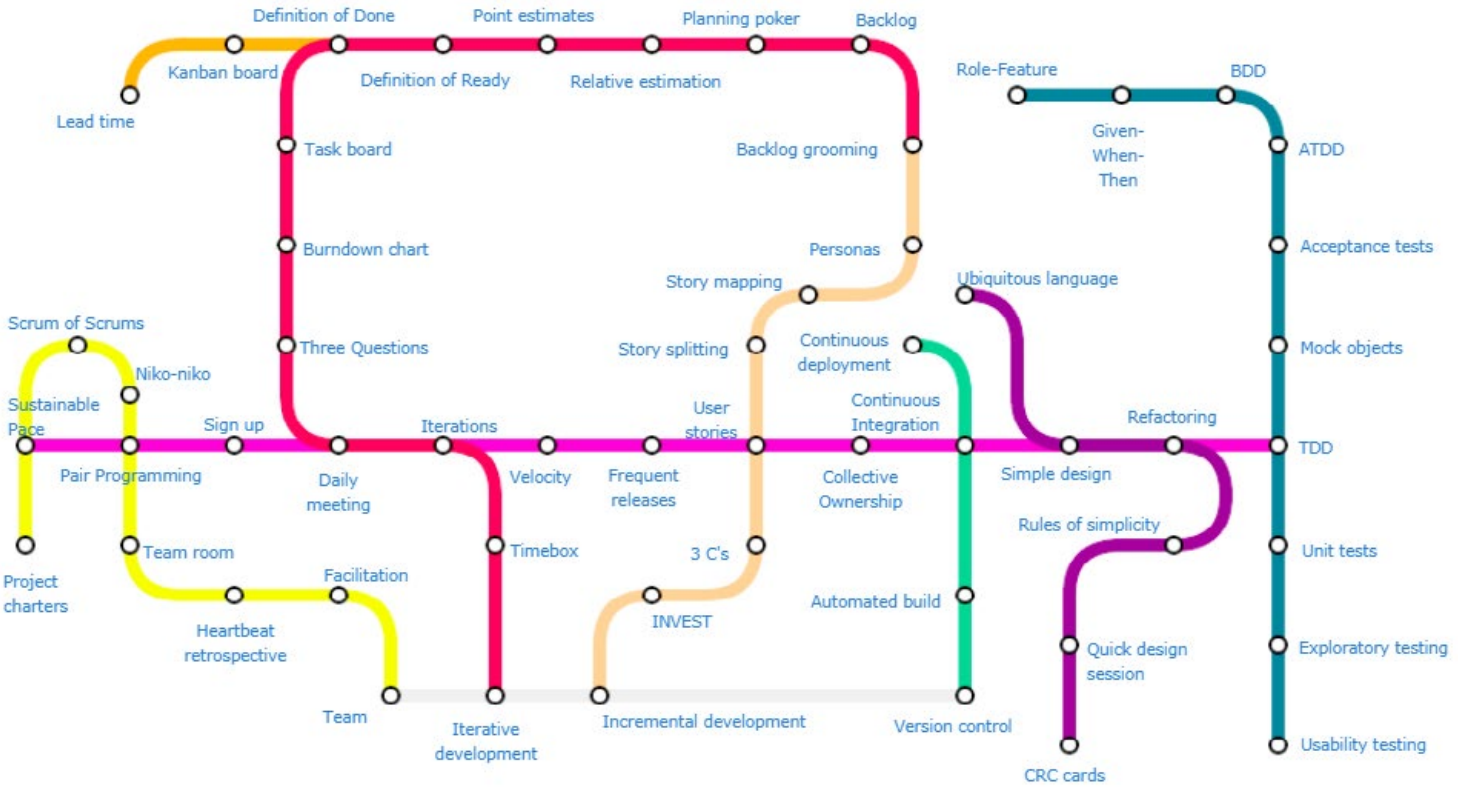
- Extreme Programming – key practices:
- Feature Driven Development
- Kanban
- Scrum
- Lean





Why the Fact of Agile is Not Enough for Prediction

The fact that your project is agile is not a cost driver!



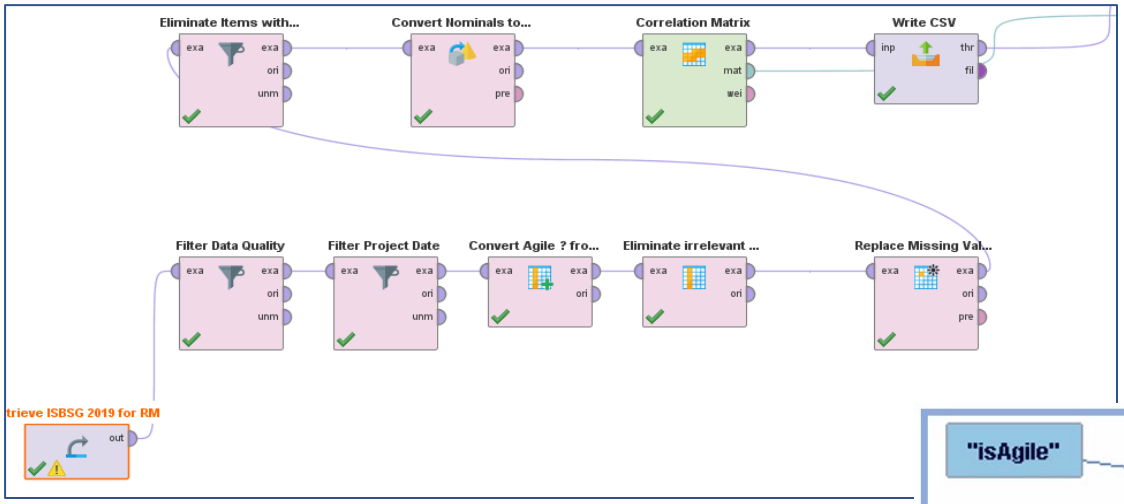
Lines represent practices from the various Agile "tribes" or areas of concern:

Extreme Programming	Scrum	Design
Teams	Product management	Testing
Lean	Devops	Fundamentals

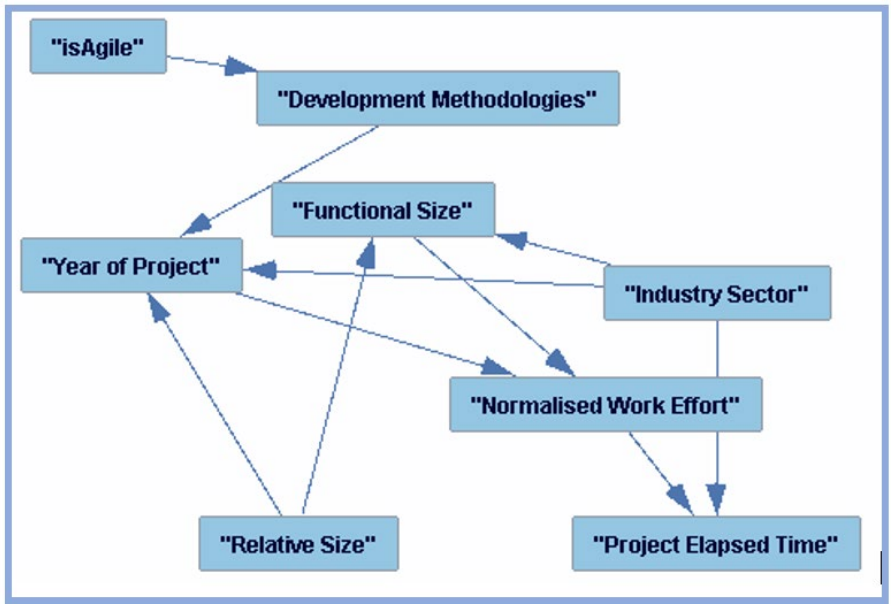
Is There Causality Between Agility and Effort

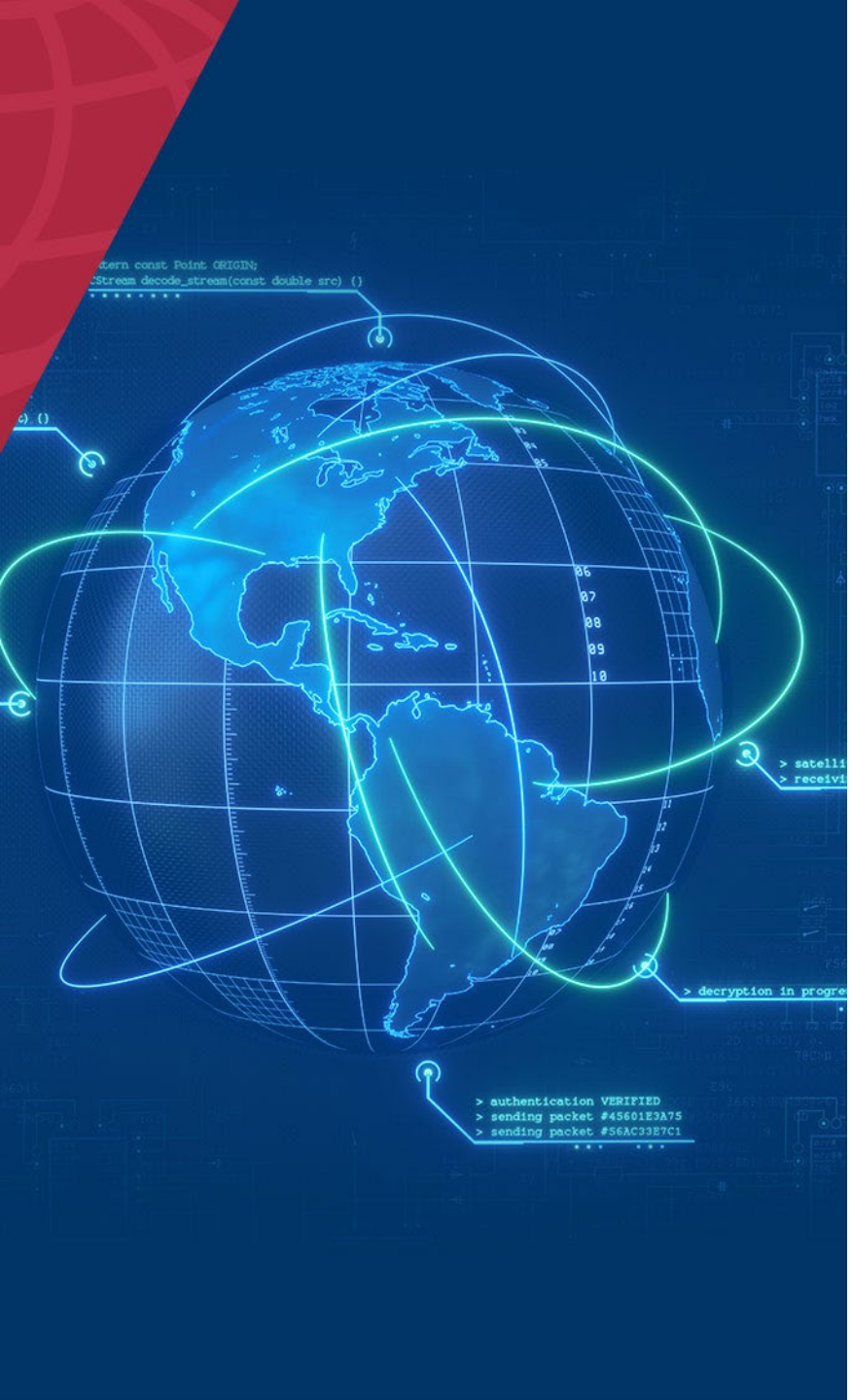
- Brief study of Data from International Benchmark Standards Group (ISBSG) – Development & Enhancement Data Base
- Started with 9100 data points. Massaged as follows:
  - Quality Rating = A or B
  - Project start year 2009 or greater (more than 99% that reported agile)
  - Normalized effort and Functional Size >0
  - Convert Agile Methods = Yes to 1, Agile Methods = No or blank to 0
- Used RapidMiner to Process Data – making it possible to complete various excursions.
- Rapid Miner created a CSV file
- Fed this CSV file to Tetrad for to perform a causal analysis

Causal Link Between Agile-ness and Effort



Industry Sector	Non Agile	Agile
All	3181	345
Banking	85	29
Education	31	13
Government	223	86
Insurance	226	181
Professional Services	33	13
Utilities	14	5





Methodologies and Rules of Thumb for Agile Estimation

Estimating Agile Software Projects

- Conundrum around all software estimates – when the first estimate is required – information is incomplete
- Good understanding of project, technical and team parameters is paramount to credible estimates
- When the development team declares agility – how is the estimator to respond
  - Is the team truly agile – then the estimate is bounded by team size, number of iterations and schedule
  - Most truly agile teams are not quite this agile – especially in the government space – the program is still likely held to some or all of the following
    - *Schedule Constraints*
    - *Firm fixed price*
    - *Relatively fixed requirements (at least for the Minimal Viable Product (MVP)*
  - This second category of agile teams are those who haven chosen to use agile practices within the constraints of government contracts and acquisition mandates

Agile Estimation

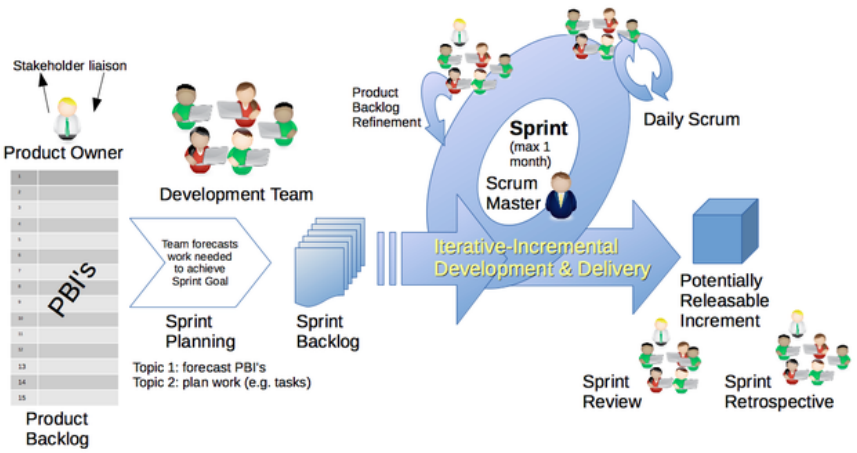
- How to adapt estimation methodology within this wide spectrum of agile possibilities
- At the far left – the truly agile team.
  - What is known
    - *Team Size*
    - *Length of iterations or sprints*
    - *Expected deliver date*
    - *Expected release cadence*
    - *Agile practices employed (conversation between agile team and developers)*
  - The estimators job becomes
    - *Apply their estimation methodology to what is known about the expected schedule and capability*
    - *Take into account affects of specific agile practices*
    - *Offer agile team and decision makers advice about schedule feasibility and risk of delivering expected capability*
- Significant variance between to two estimation methods should stimulate discussion leading to better convergance

Agile Estimation

- How to adapt estimation methodology within this wide spectrum of agile possibilities
- At the other end of the spectrum
  - What is known
    - *Capability requirements*
    - *Schedule constraints*
    - *Agile practices employed (conversation between agile team and developers)*
  - The estimators job is not different than a traditional estimate
    - *Apply their estimation methodology to what is known about the expected schedule and capability*
    - *Take into account affects of specific agile practices*
    - *Create an cost, effort and schedule estimate*
- The advantage in this scenario is that agile team should have collected significant metrics against which this traditional method results can be compared – also facilitating discussions that should lead to convergence

Agile Cost Drivers

- The fact that your project is agile is not a cost driver
- There are potential cost implications to adopting agile practice
- Estimation team needs to determine which agile practices apply and how they impact cost and schedule



Agile Cost Drivers

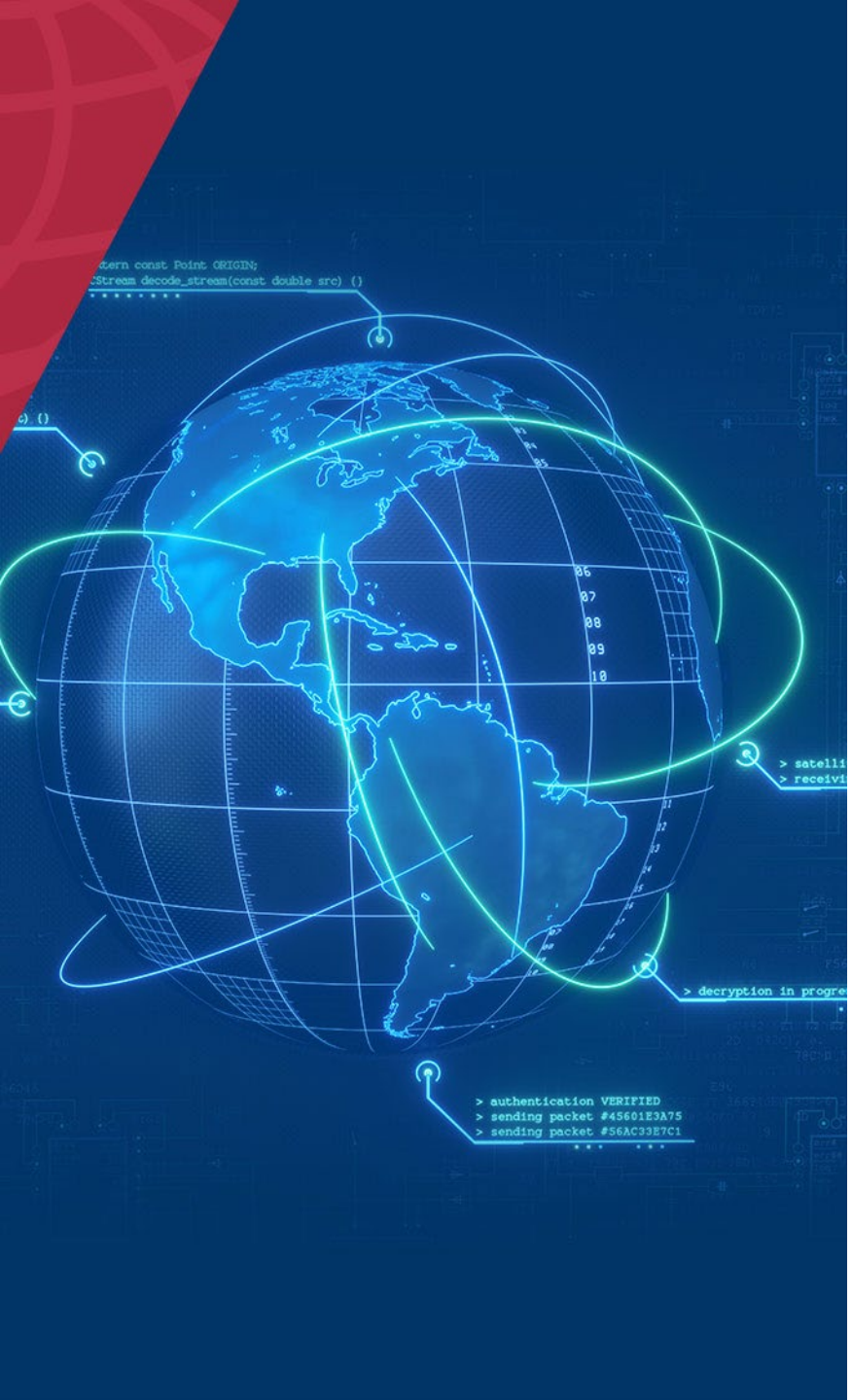
- Agile teams tend to be highly skilled
  - *Hard to be a slacker in an agile environment*
  - *Working closely with high skilled team members, learning curve for new members is quick*
  - *Input parameters to your model indicating team experience would be affected*
- Agile teams tend to have tool sets that are quite sophisticated
  - *This would be especially true on teams working with space systems as it would greatly facilitate compliance to standards*
  - *Input parameters around tools or automation would be affected*

Agile Cost Drivers

- Co-location of teams should improve team productivity
  - *Culture of interruption*
  - *Questions answered in real time*
  - *Team cohesion increases*
  - *Co-locating stakeholders and SMEs with development team creates a real time IPT*
  - *Well run stand-up meetings increase productivity and quality*
  - *Cost drivers indicating distribution of team and communication practices would be affected*

Agile Cost Drivers

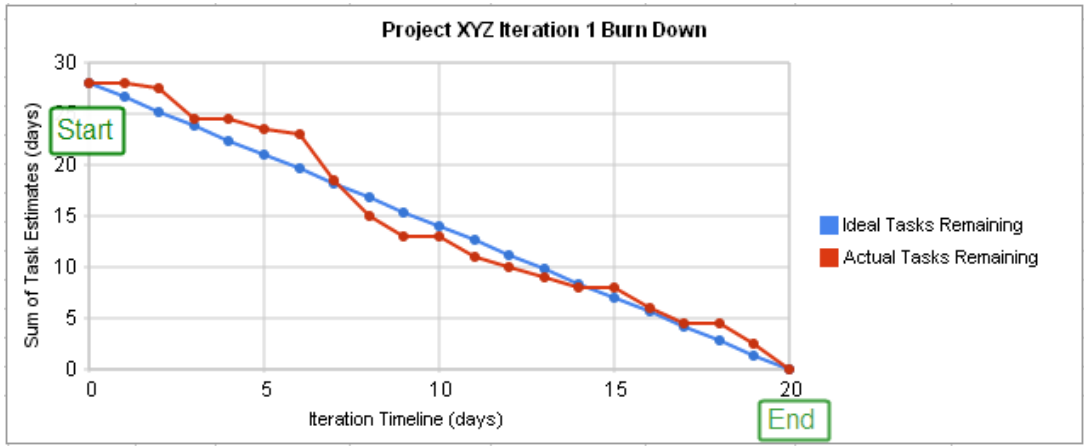
- Continuous integration with automated testing should increase delivery productivity
  - *Important in space systems to maintain safety critical compliance requirements.*
  - *Code is checked in frequently and builds are run and test regularly before developers forget what they changed*
  - *Red tests raise red flags – team fixes them right away*
  - *Since little code is changed, errors are easy to track down*
  - *Fixes occur quickly*
  - *Cost drivers focused on integration test complexity would be affected*



Useful Agile Metrics

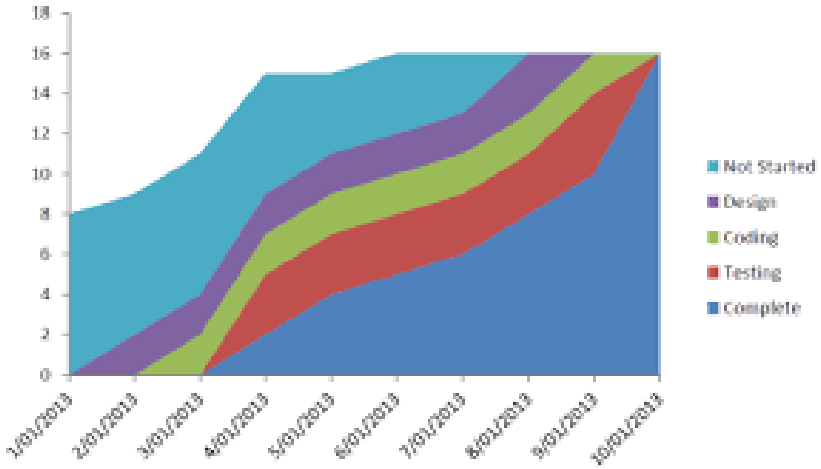
Team Metrics

- Many agile teams have retrospectives at each iteration or sprint
- Agile team metrics are team specific
- Common team metrics include:
  - Stories or story points per iteration
  - Defects inject/removed during iteration
  - Velocity
  - Burn down chart
  - Burn up chart
  - Cycle time



Project Control Metrics

- Team metrics are not good for project control in cases where there are multiple teams across a project
- Metrics can be normalized across teams for project level
- Common Project Control Metrics include:
  - Epic/Release Burn Up Chart
  - Epic/Release Burn Down Chart
  - Product Backlog
  - Changes to Product Backlog
  - Defects injected
  - Defects Removed
  - Latent Defects Delivered to the field
  - Cumulative Flow Diagram
  - Number of Features Delivered



Wrap Up

- The Agile Paradigm is here to stay
- Current practices of agile, particularly in the government, are limited, not necessarily truly agile
  - Entirely reasonable approach, enabling the benefits of agile without risking consequences of failed contracts or unmet expectations
- Estimators need to understand what ‘agile’ means:
  - In the context of the project that is being estimated
  - In the context of the agile team(s) working on the project
  - In the context of the agile practices being utilized on the project
- The fact that your project is agile is not a cost driver, but it should start a conversation about how agile practices may influence your cost estsimate

Questions?