

Get to CMMI ML3 Using Agile Development Processes for Large Projects

Catherine Clark, Business Solutions Architect



Can CMMI and Agile complement each other?

Can CMMI and Agile complement each other?

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.

Through this work 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.

Can CMMI and Agile complement each other?

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.

Through this work 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.

Can CMMI and Agile complement each other?

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.

Through this work 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.

Can CMMI and Agile complement each other?

Manifesto for Agile Software Development

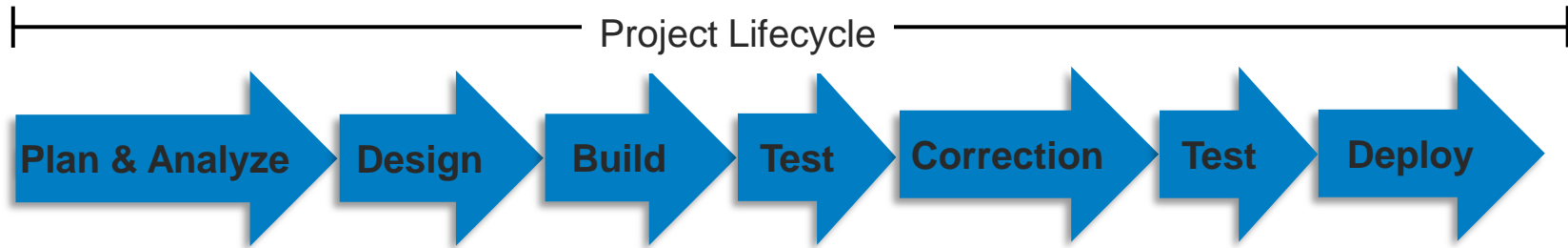
We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

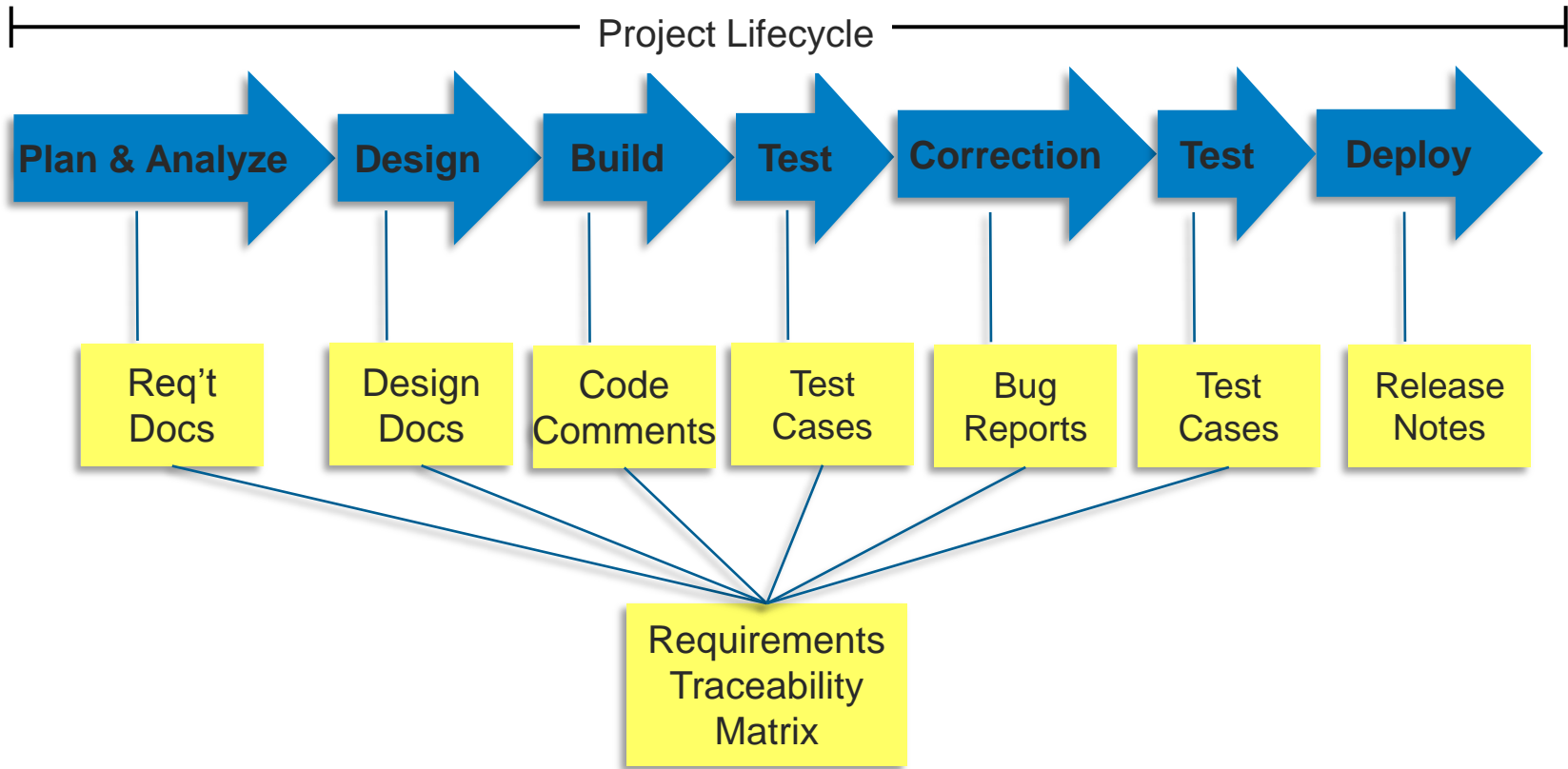
Individuals and interactions over **processes** and **tools**
Working software over **comprehensive documentation**
Customer collaboration over contract negotiation
Responding to change over following a **plan**

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

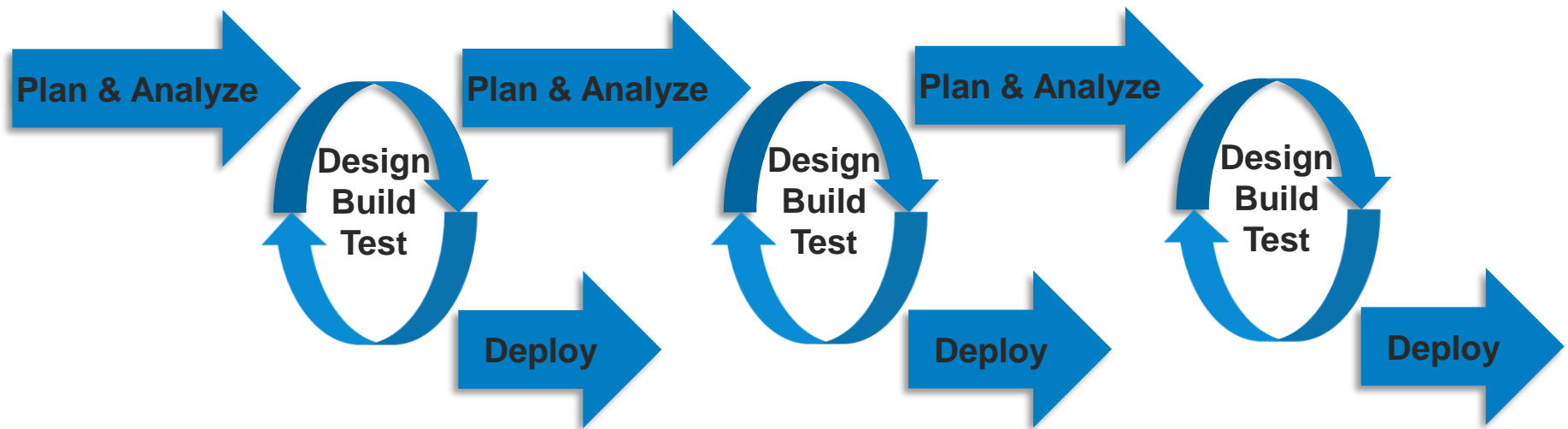
Waterfall Method



Waterfall Method

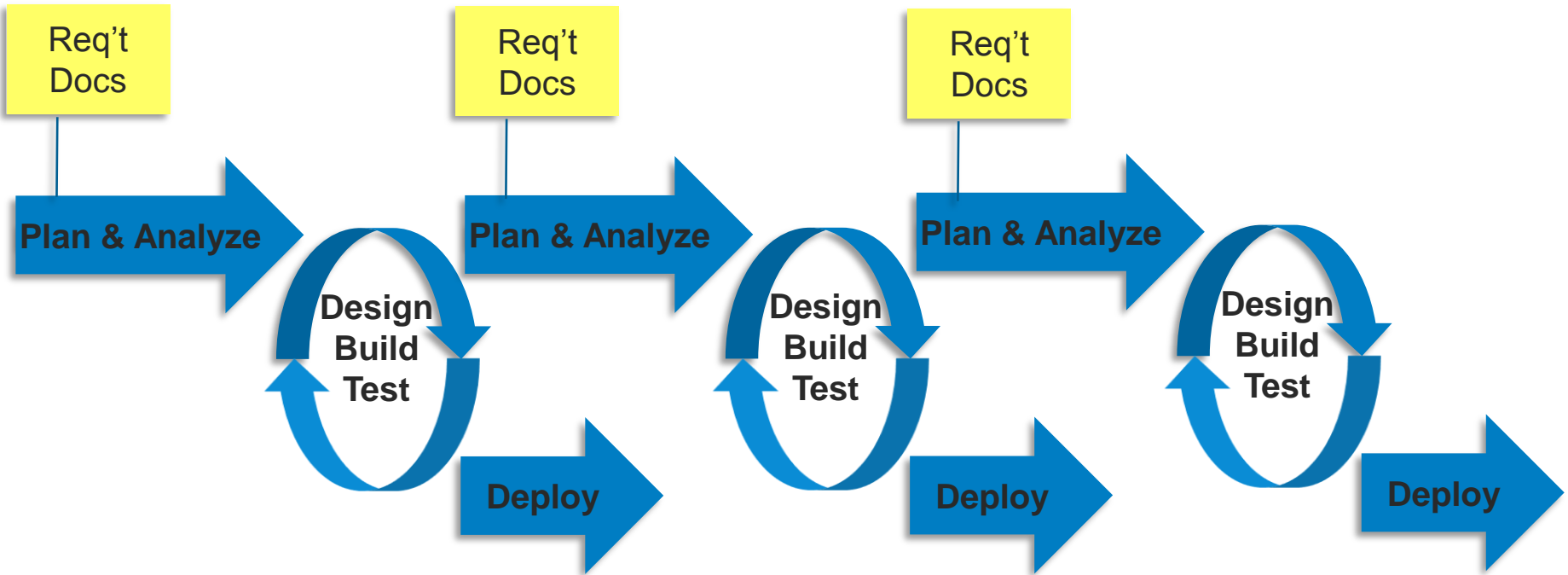


Agile Method



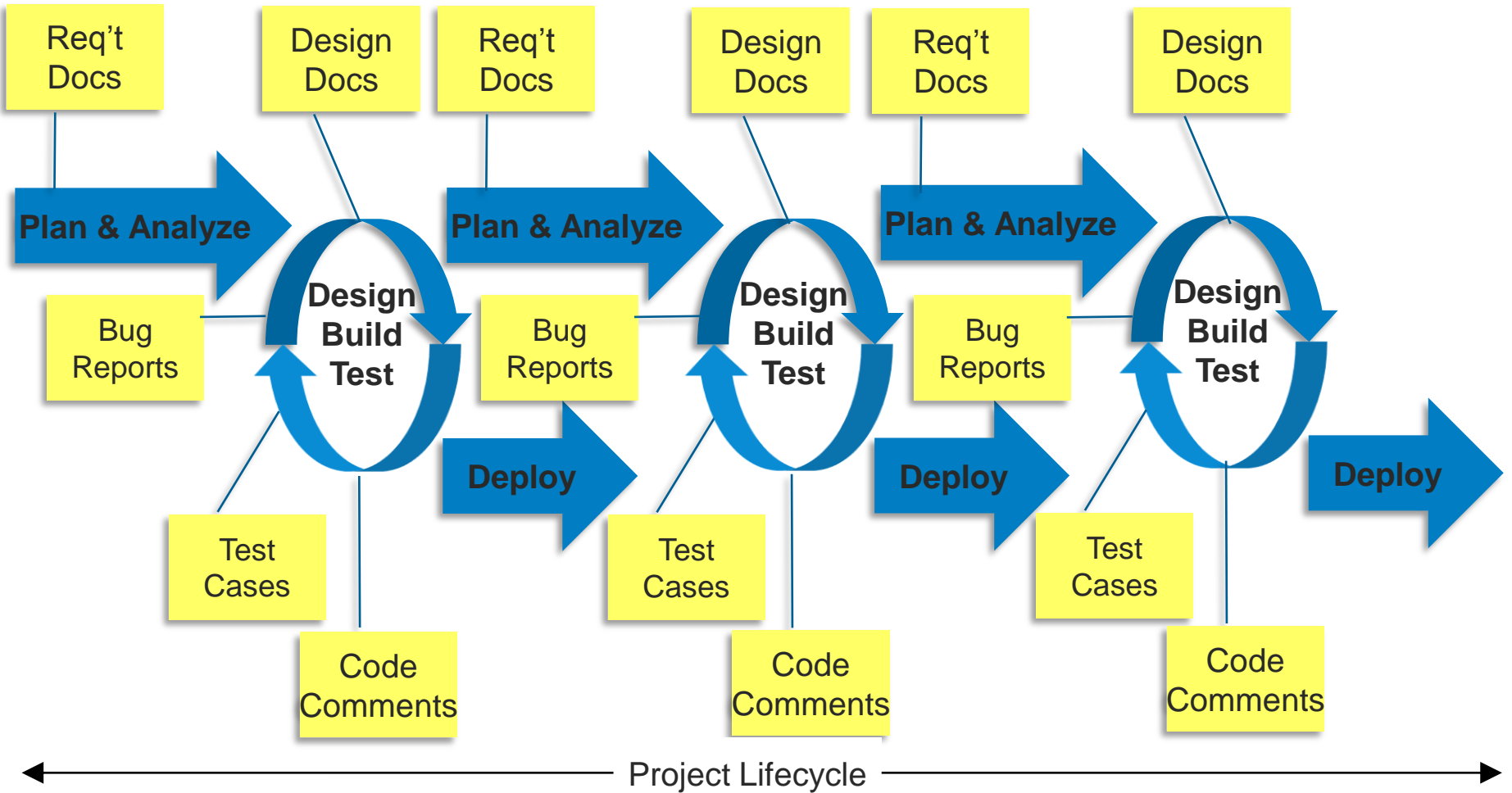
← Project Lifecycle →

Agile Method

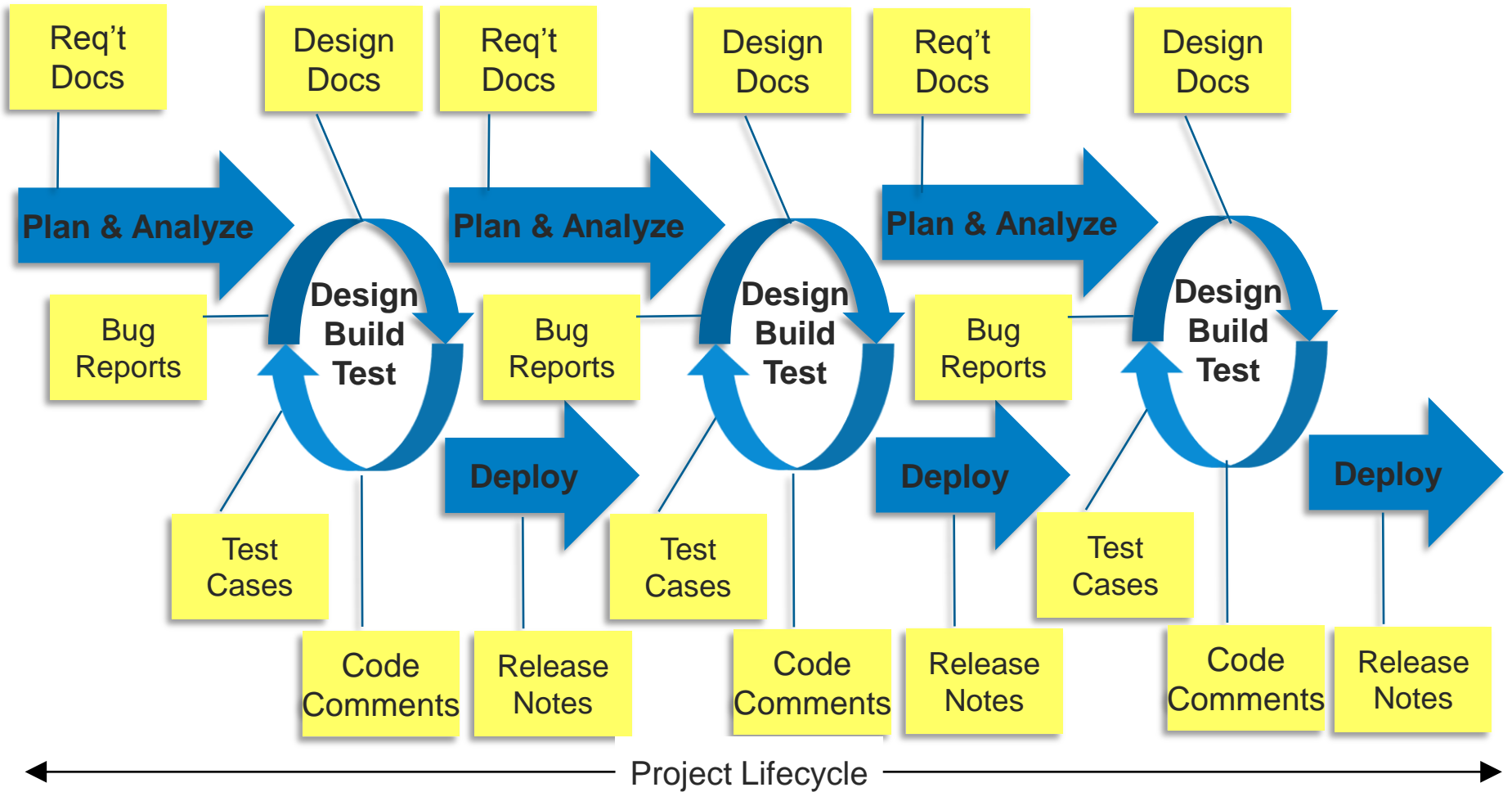


← Project Lifecycle →

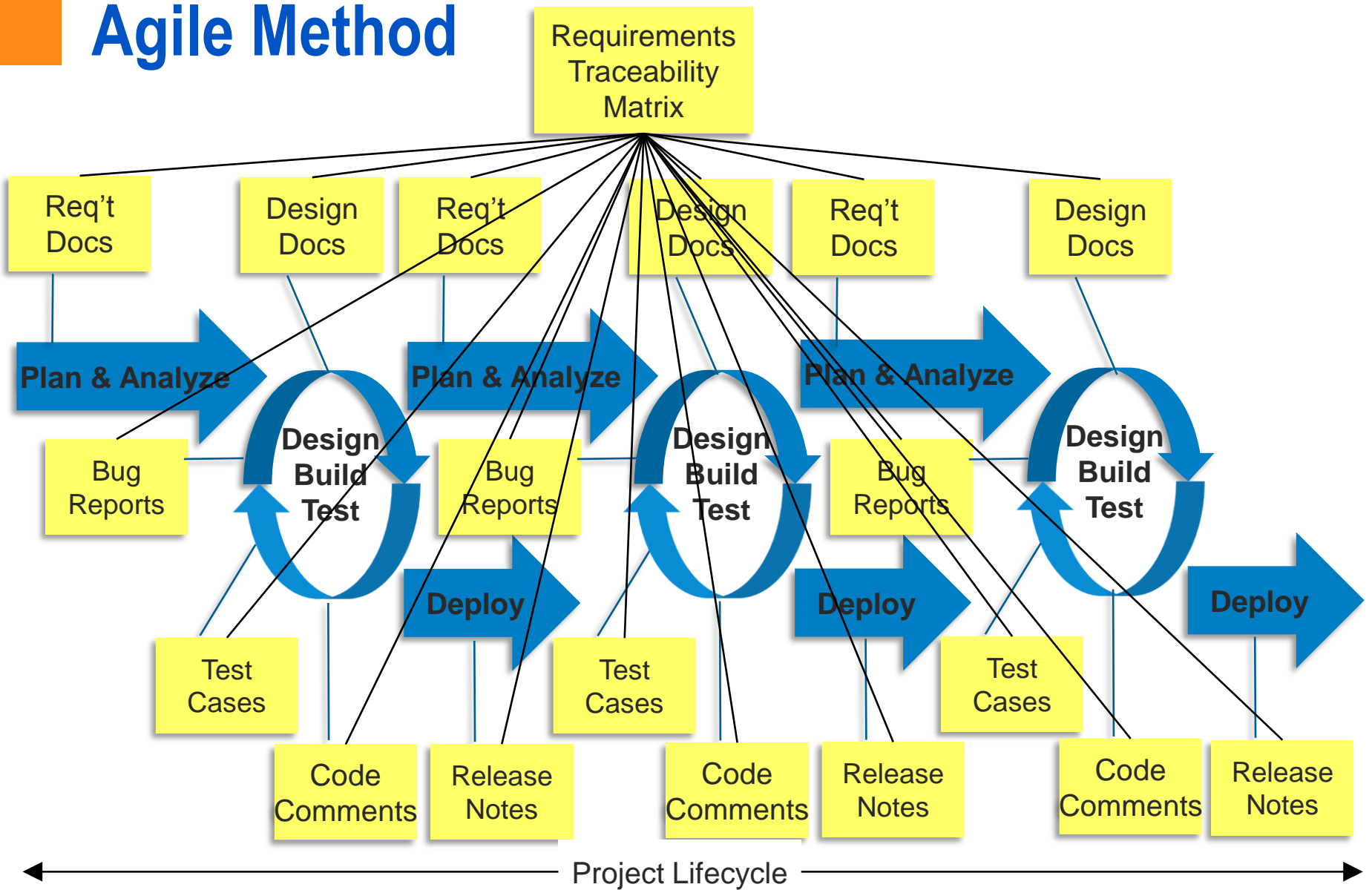
Agile Method



Agile Method



Agile Method



Can CMMI and Agile complement each other?

- How can this be achieved while still being Agile and keeping the developers focused on:

Individuals and interactions

Working software

Customer collaboration

Responding to change

Two parts of the CMMI appraisal process

- Evidence
 - Direct
 - Indirect
- Affirmations/Interviews

Two parts of the CMMI appraisal process

- Evidence
- **Affirmations/Interviews**

Affirmations/Interviews

- Interviews are fashioned to fill in any blanks that the evidence didn't show, or was unclear
- **Daily scrums are critical** in ensuring all of the teams are aware of the process

Two parts of the appraisal process

- Evidence
- Interviews

How do you generate evidence while keeping the team actively engaged in producing working software?

- Use the tools they are already using every day...



SUBVERSION®



Jenkins



How do you generate evidence while keeping the team actively engaged in producing working software?

- Use the tools they are already using every day and adjust and integrate them to maintain the documentation required for CMMI evidence



Agile/CMMI Process

Req't
Docs



User Stories and Tasks are kept in Rally.

Agile/CMMI Process

Req't
Docs



User Stories and Tasks are kept in Rally.

Code
Comments



Code and Comments are written in Eclipse and stored via Subversion. When committing, Rally will store a link to the Code.

Agile/CMMI Process

Req't
Docs



User Stories and Tasks are kept in Rally.

Code
Comments



Code and Comments are written in Eclipse and stored via Subversion. When committing, Rally will store a link to the Code.

Test
Cases



Automated Test Cases reside in Jenkins, Rally can integrate and correlate them to the User Stories

Agile/CMMI Process

Req't
Docs



User Stories and Tasks are kept in Rally.

Code
Comments



Code and Comments are written in Eclipse and stored via Subversion. When committing, Rally will store a link to the Code.

Test
Cases



Automated Test Cases reside in Jenkins, Rally can integrate and correlate them to the User Stories

Bug
Reports



Defects are kept in Rally, and can point back to Test Cases in Jenkins.

Agile/CMMI Process

Req't
Docs



User Stories and Tasks are kept in Rally.

Code
Comments



Code and Comments are written in Eclipse and stored via Subversion. When committing, Rally will store a link to the Code.

Test
Cases



Automated Test Cases reside in Jenkins, Rally can integrate and correlate them to the User Stories

Bug
Reports



Defects are kept in Rally, and can point back to Test Cases in Jenkins.

Release
Notes



Release notes are stored within the Release in Rally. Each User Story points to the code affected, as well as the Defects.

Agile/CMMI Process

Req't
Docs



User Stories and Tasks are kept in Rally.

Code
Comments



Code and Comments are written in Eclipse and stored via Subversion. When committing, Rally will store a link to the Code.

Test
Cases



Automated Test Cases reside in Jenkins, Rally can integrate and correlate them to the User Stories

Bug
Reports



Defects are kept in Rally, and can point back to Test Cases in Jenkins.

Release
Notes



Release notes are stored within the Release in Rally. Each User Story points to the code affected, as well as the Defects.

Requirements
Traceability
Matrix



The integration of these four tools allow a quick and easy way to trace from Requirement (User Story) to Code and Comments, to Defects and Test Cases, and to the Release.

Agile/CMMI Process

Req't
Docs



User Stories and Tasks are kept in Rally.

Design
Docs



Code
Comments



Code and Comments are written in Eclipse and stored via Subversion. When committing, Rally will store a link to the Code.

Test
Cases



Automated Test Cases reside in Jenkins, Rally can integrate and correlate them to the User Stories

Bug
Reports



Defects are kept in Rally, and can point back to Test Cases in Jenkins.

Release
Notes



Release notes are stored within the Release in Rally. Each User Story points to the code affected, as well as the Defects.

Requirements
Traceability
Matrix



The integration of these four tools allow a quick and easy way to trace from Requirement (User Story) to Code and Comments, to Defects and Test Cases, and to the Release.



Questions?

Catherine Clark
Business Solutions Architect
410-418-5555
Cathy.Clark@vsticorp.com

www.vsticorp.com