

INTRODUCTION

Welcome



# ADC Design Controls for Agile Software Development

CLICK THE FORWARD  
ARROW TO BEGIN.



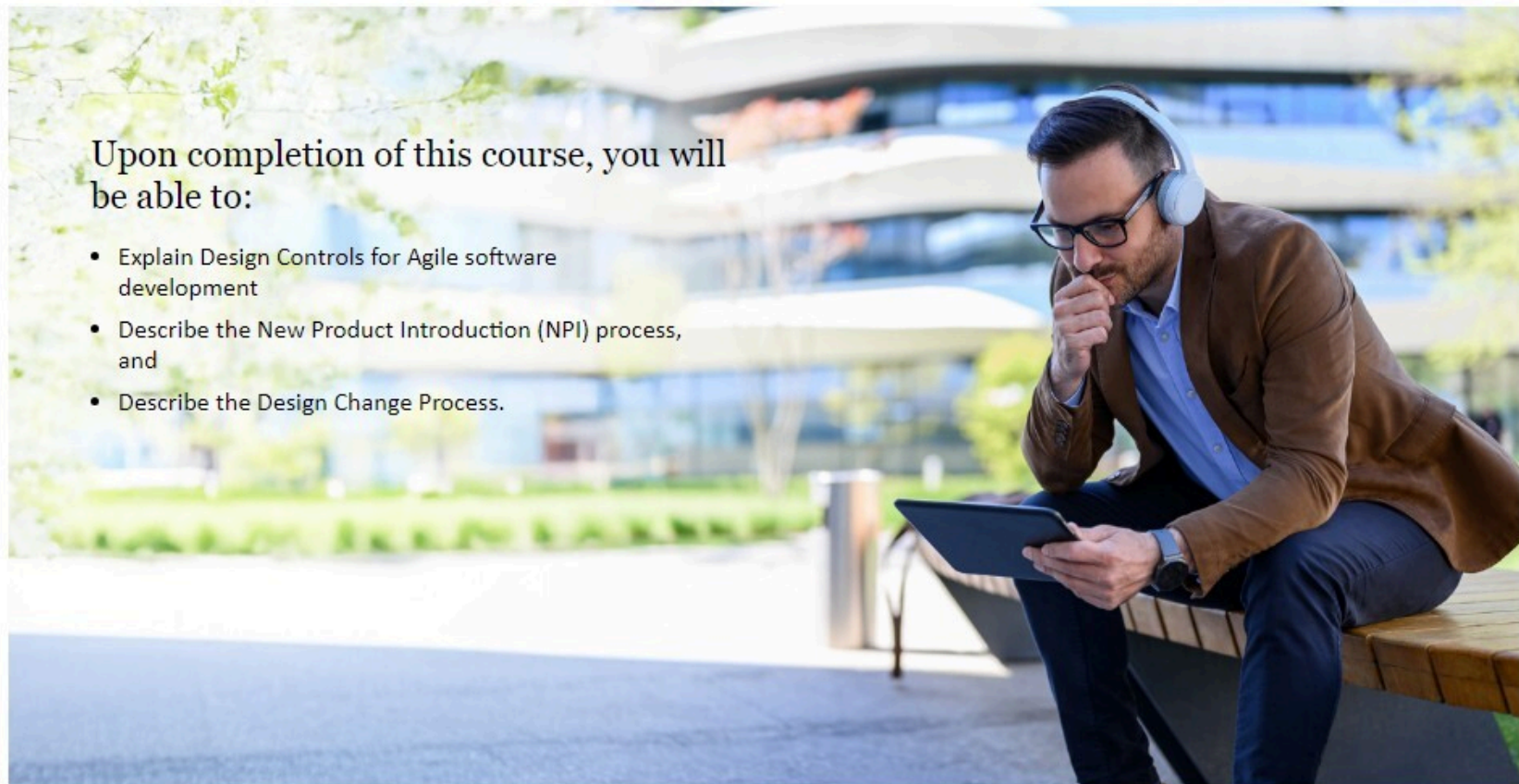
## INTRODUCTION

# Objectives




Upon completion of this course, you will be able to:

- Explain Design Controls for Agile software development
- Describe the New Product Introduction (NPI) process, and
- Describe the Design Change Process.






# Menu



## 1 | Understanding Agile Software Development

Here you will learn what is Agile software development.

4 MINUTES 

Click the panel to get started.

## Controls



Here you will learn about ADC's Agile design control process.

8 MINUTES 



## 3 | Knowledge Check

Assess your understanding of the key concepts and principles of this course.

3 MINUTES 





UNDERSTANDING AGILE SOFTWARE DEVELOPMENT

# What is Agile Software Development?



At Abbott Diabetes Care (ADC), we use an Agile software development framework for the development of medical devices.



# What is Agile Software Development?



The ADC Agile design controls framework is based on the FDA AAMI TIR45:2023 guidance on the use of Agile practices in development of medical devices software.

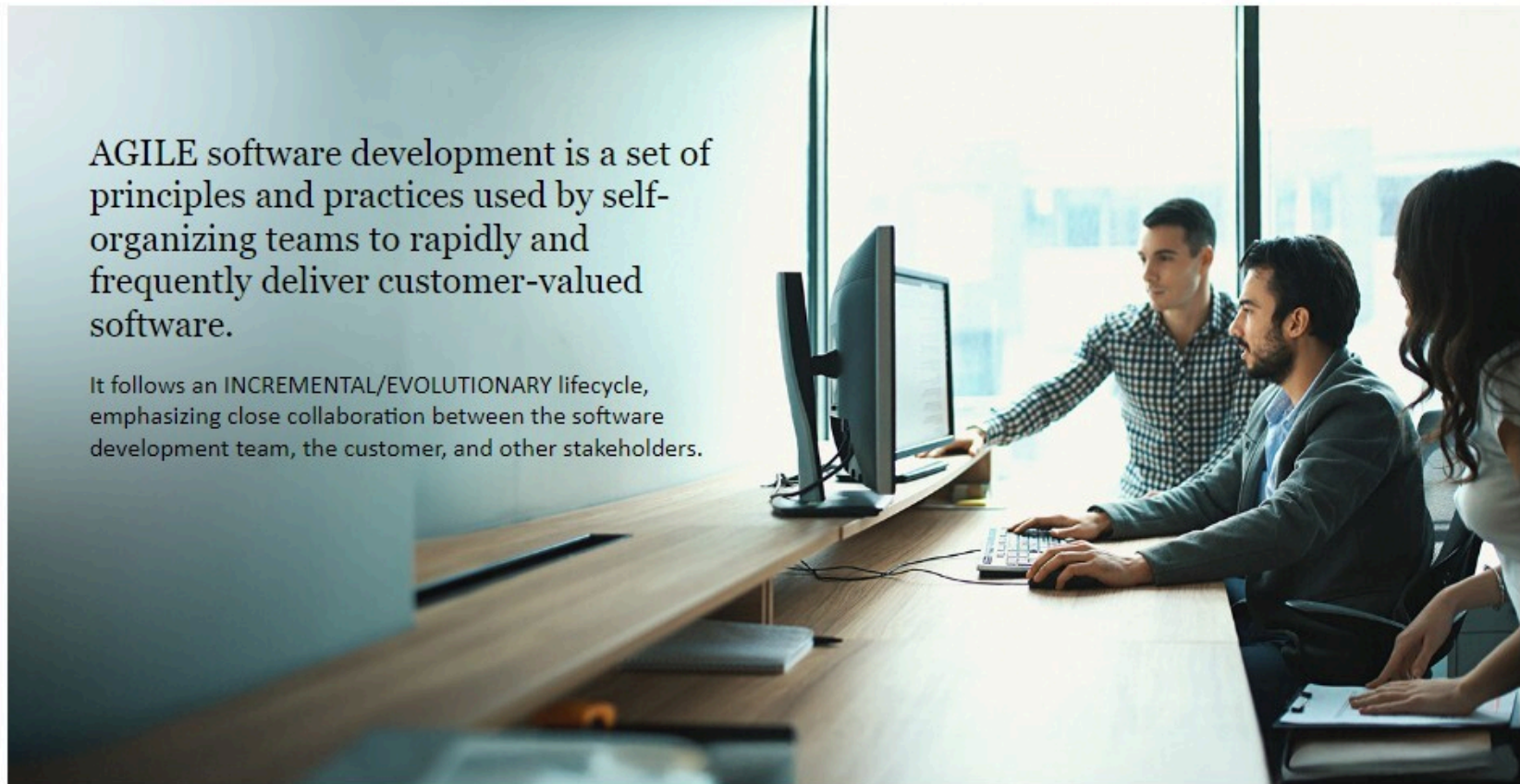


# What is Agile Software Development?



AGILE software development is a set of principles and practices used by self-organizing teams to rapidly and frequently deliver customer-valued software.

It follows an INCREMENTAL/EVOLUTIONARY lifecycle, emphasizing close collaboration between the software development team, the customer, and other stakeholders.



# What is Agile Software Development?



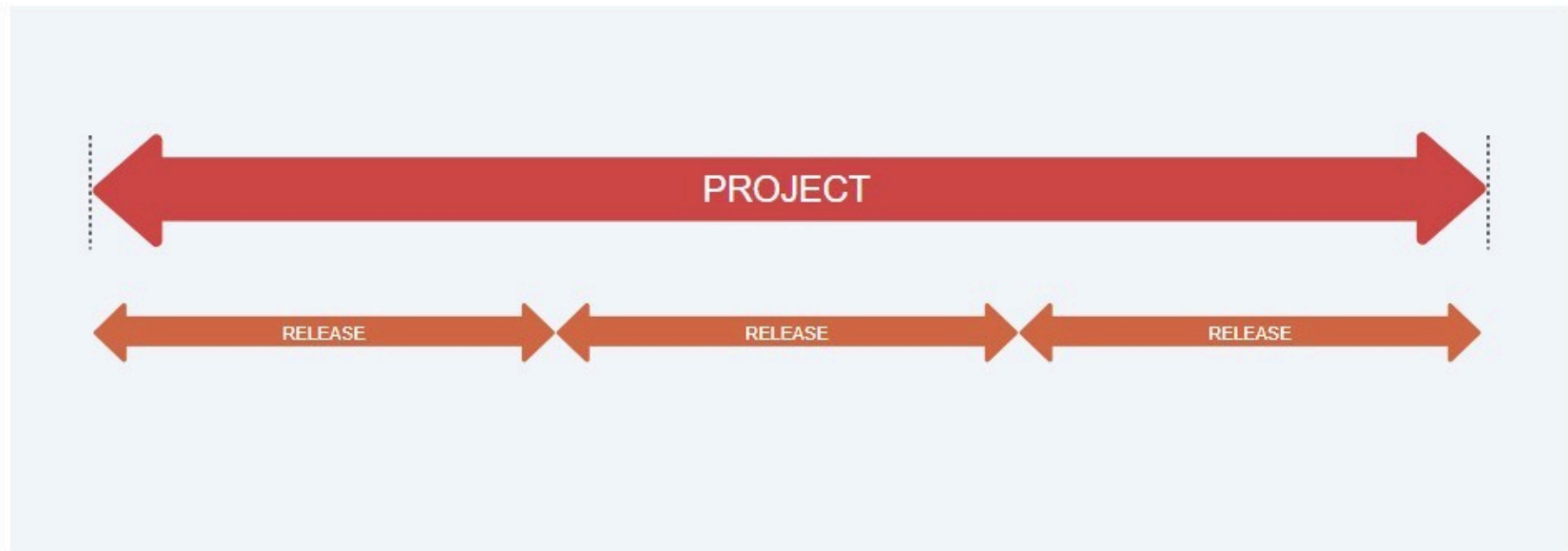
Let's take a closer look at the process.

In AGILE software development, each project is developed iteratively and incrementally.





# What is Agile Software Development?



Typically, a project (or product) has multiple releases.





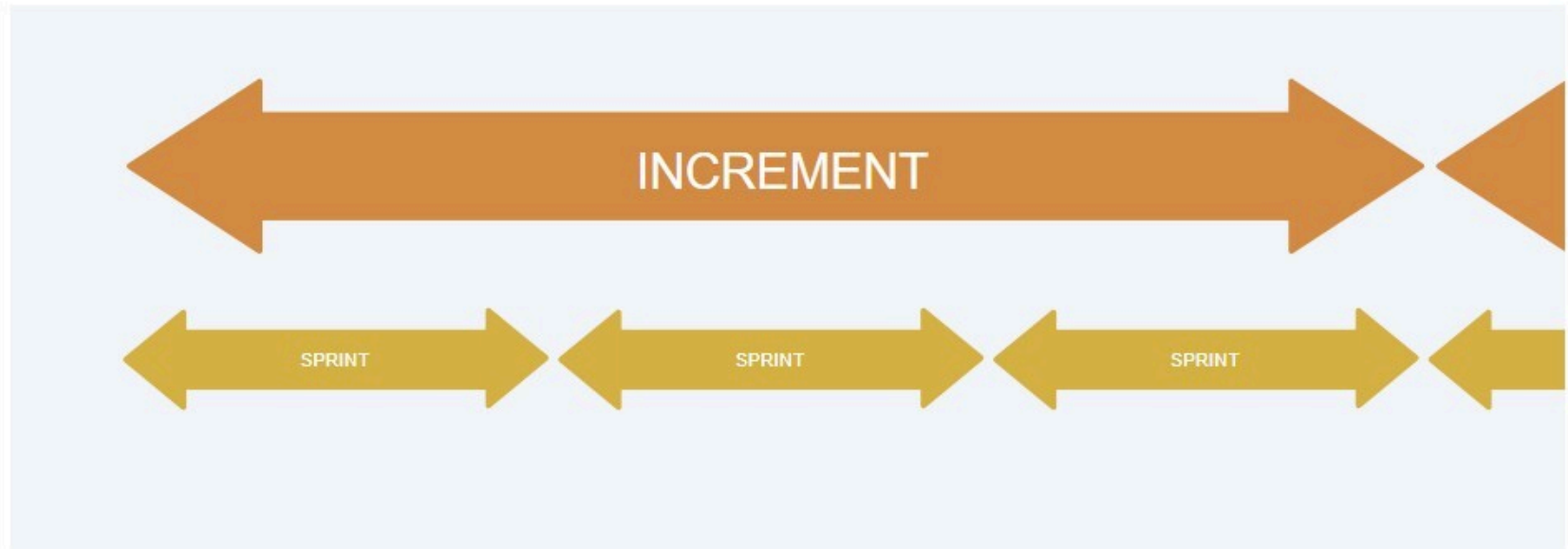
# What is Agile Software Development?



Each release is broken into segments called increments.



# What is Agile Software Development?

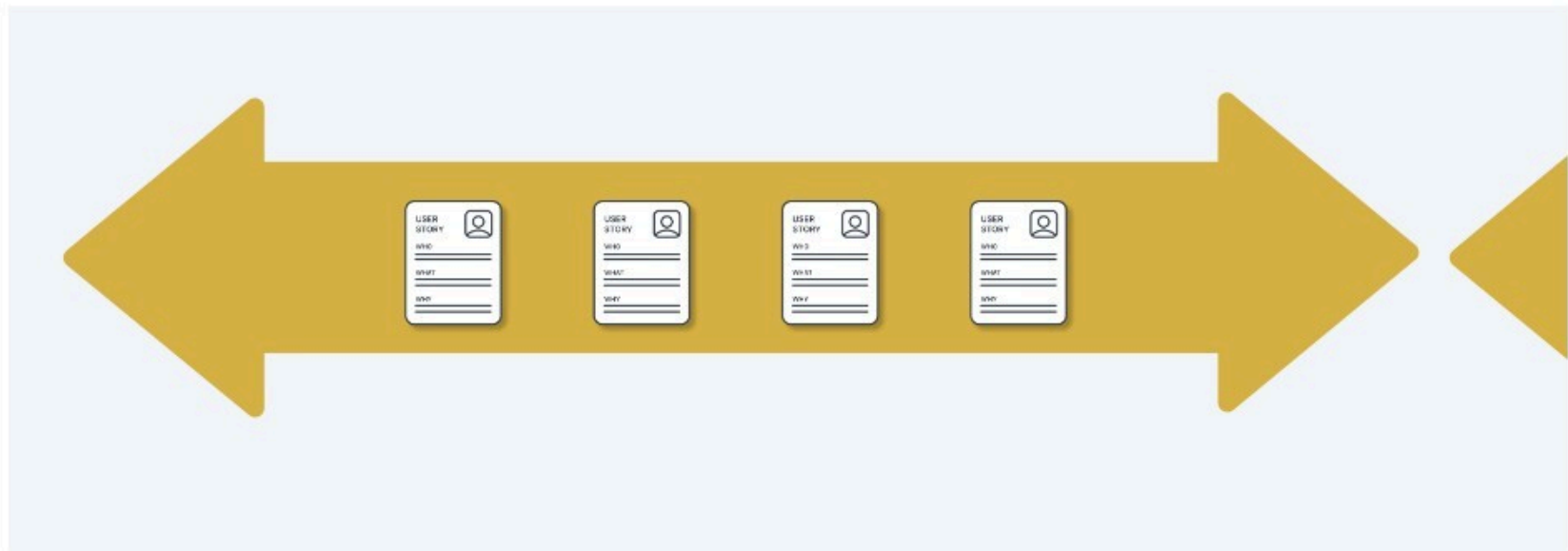


These increments are further broken down into shorter cycles called sprints, which typically are “time-boxed” to last 1-2 weeks.



## UNDERSTANDING AGILE SOFTWARE DEVELOPMENT

# What is Agile Software Development?



Sprints are limited in scope and focus on the development of a small number of features or tasks also known as stories.



# What is Agile Software Development?



USER STORY



WHO

WHAT

WHY

USER STORY



WHO

WHAT

WHY

USER STORY



WHO

WHAT

WHY

A user STORY is a short, simple description of a feature told from the perspective of the person who desires the new capability, usually a user or customer of the system.

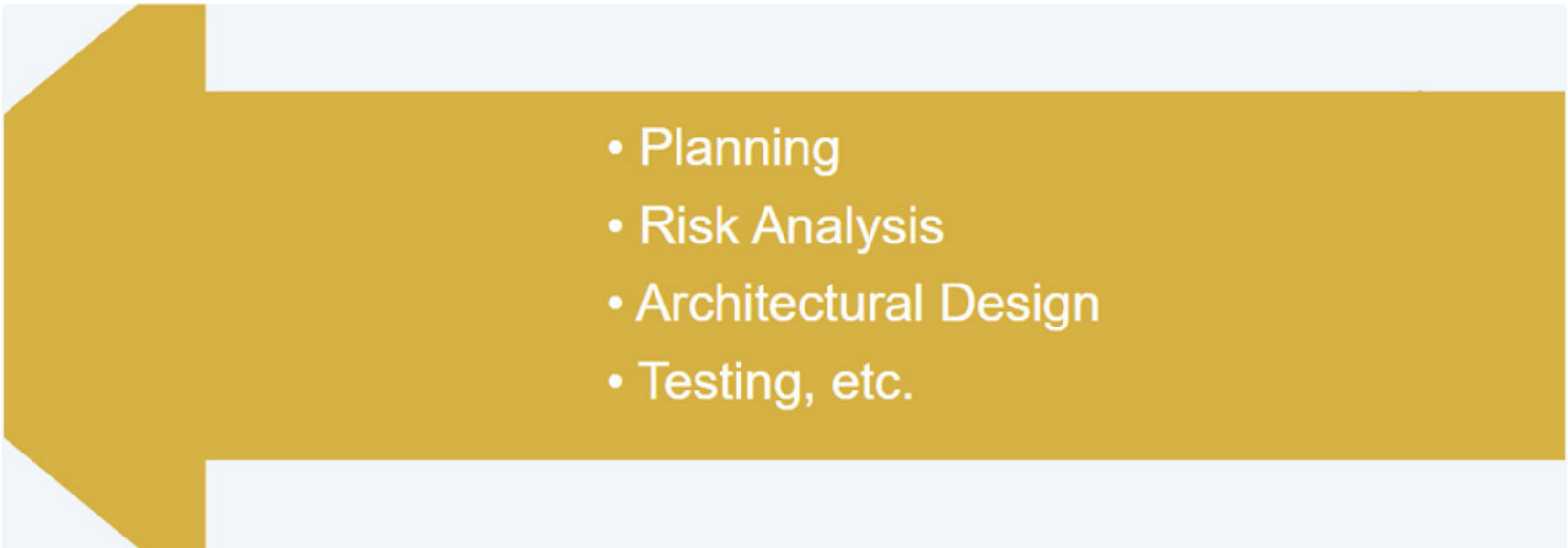
They typically follow a simple who/what/why template. STORIES typically include accompanying acceptance criteria.





# What is Agile Software Development?



- 
- Planning
  - Risk Analysis
  - Architectural Design
  - Testing, etc.

Each level of development has a number of activities associated with it.

For example, Sprints typically include planning, requirements development, risk analysis, architecture design, detailed design and testing iteratively until the increment or release scope is complete.



## UNDERSTANDING AGILE SOFTWARE DEVELOPMENT

# What is Agile Software Development?



## For Each Product

- Design & Development Planning – Establish planning & execution strategies
- Requirement – Backlog items contain initial definition, spikes for intentional definition, guidance to teams for emergent definition
- Risk Management, FMEA – Backlog items and DoD addressing software-specific activities
- Architectural Design – Backlog Structure, spikes for intentional architecture, guidance to teams for emergent architecture

## For Each Release

- Change Planning – Release Plan (scope and purpose, increment strategy, backlog sorting/sequencing, debt management, DoD)

## For Each Iteration and/or Increment

- Change Planning – Increment Plan (scope and purpose, backlog sorting/sequencing, debt management, DoD)

## For Each Backlog Item (Stories) (Spikes)

- Requirements – Definition Spikes
- Architecture & Design – Solution Spikes

## (Changes)

- Planning – Define activities, documentation, DoD
- Risk Management, FMEA – Impact of Change
- Requirements – Create/modify, trace
- Architecture – Create/modify
- Detailed Design – Create/modify
- Unit – Create code & unit tests, execute tests
- Dry run Verification or Design Verification Testing (DVT)
- SW & Sys Integr Test – Create tests, execute (if able)
- Validation – PO Acceptance, Story-specific val activities, Demo
- Defect Management (on-market product assessment)

Other activities as applicable to the product/system...

Req – (Sub)set generated, approved

Risk Management, FMEA – Aggregate Review

Arch – (Sub)set generated, approved

Det Des – (perhaps do nothing here)

Unit Test – Regression, Increment Report

Design Verification Testing (DVT)

SW & System Integration Testing

Defect Management, Fix or defer defects

Formal Design reviews

Validation – Increment demo, incremental validation activities-device validation eg. human factors, clinical validation

Other activities as applicable to the product/system...

Req – Complete c  
generated, appro

Risk Management  
Management Rep

Arch – Complete

Det Des – Compli  
report generated

Unit Test – Regre

Design Verificatio

Defect Managem  
for next release

SW & Sys Integra  
run & final report

Formal Design re

Validation – Dem  
activities, final rep

Release

Other activities as applicable to the  
product/system...

The ADC Agile development framework shown on screen is an important resource.

It identifies the activities and deliverables associated with each level of development, for example, project, release, increment, story. The process to execute these activities in New Product Introduction (NPI) and design changes will be explained in the next module.





# What is Agile Software Development?



## Definition of Ready (DOR):

The compilation of all activities necessary to begin work on a STORY, INCREMENT and RELEASE.

## Definition of Done (DOD):

Establishes expectations for the activities and objective evidence for completing a STORY, INCREMENT, and RELEASE.

As we move forward, here are a few important definitions to keep in mind.





# Review



## Review

Take a moment to review some of the key concepts in this section.

Click the arrow to begin your review.







# Review



At ADC, we use an Agile software development framework for the development of medical devices.



# Review



AGILE software development is performed iteratively and incrementally.



# Review



The ADC Agile design controls guidance (7.3.200G01) identifies activities at each level of development.





UNDE

Re

To check your progress, click  
the Menu button



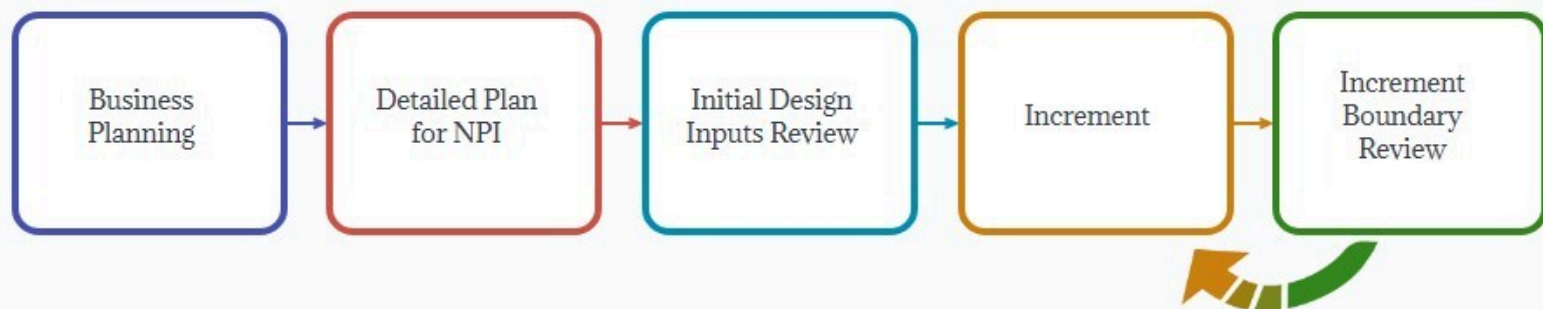
You have completed section 1 of 3

CLICK THE FORWARD ARROW TO CONTINUE LEARNING





# Agile Design Controls Process

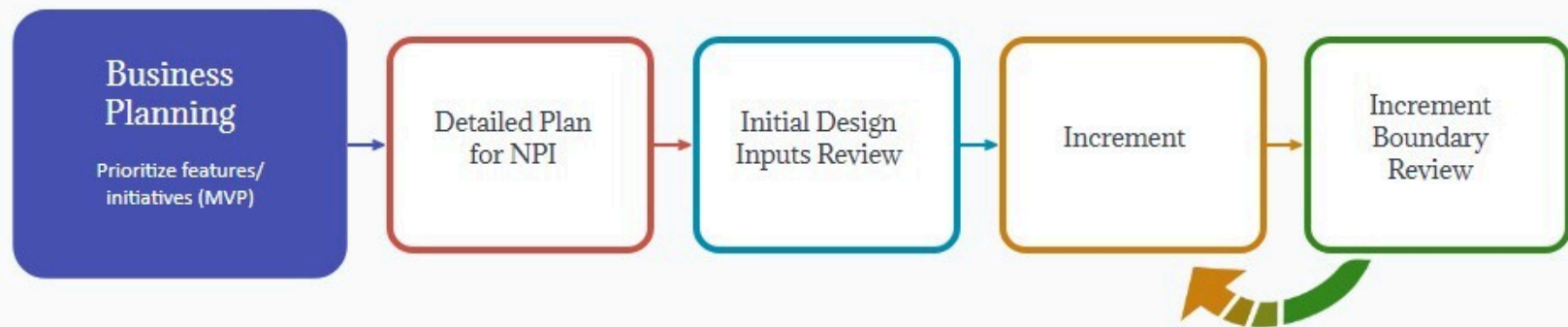


Let's take look at the Agile design controls process for New Product Introductions (NPIs).

Note that the process flow we will review over the next few screens represents a high-level overview. For more detailed information on NPI-specific process for Agile, please refer to Agile design controls software guidance document 7.3.200G01.



# Agile Design Controls Process

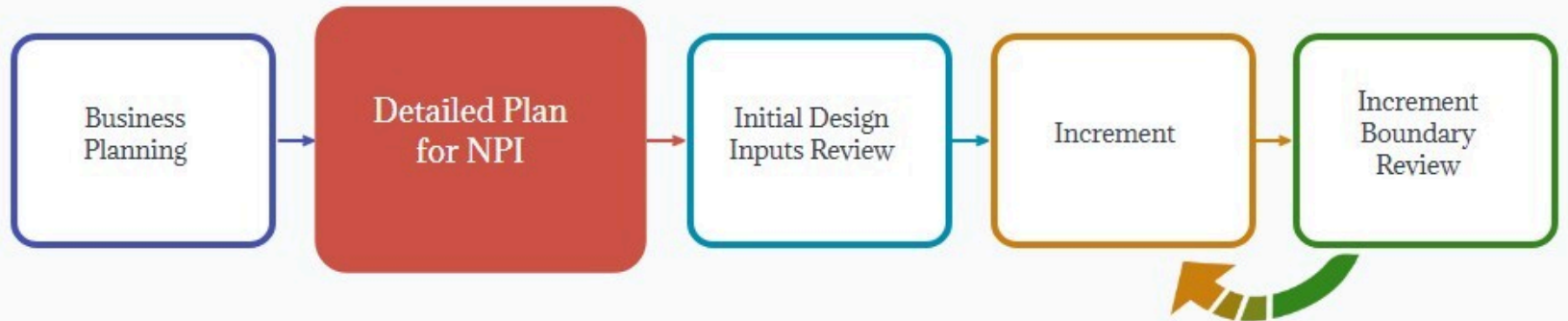


The process begins with business planning.

Representatives from Global Strategic Marketing (GSM), the Commercial Organization, and the Business PMO work alongside the R&D Program Manager to prioritize a list of features or initiatives that is planned for the software release. This is commonly known as Minimum Viable Product (MVP).



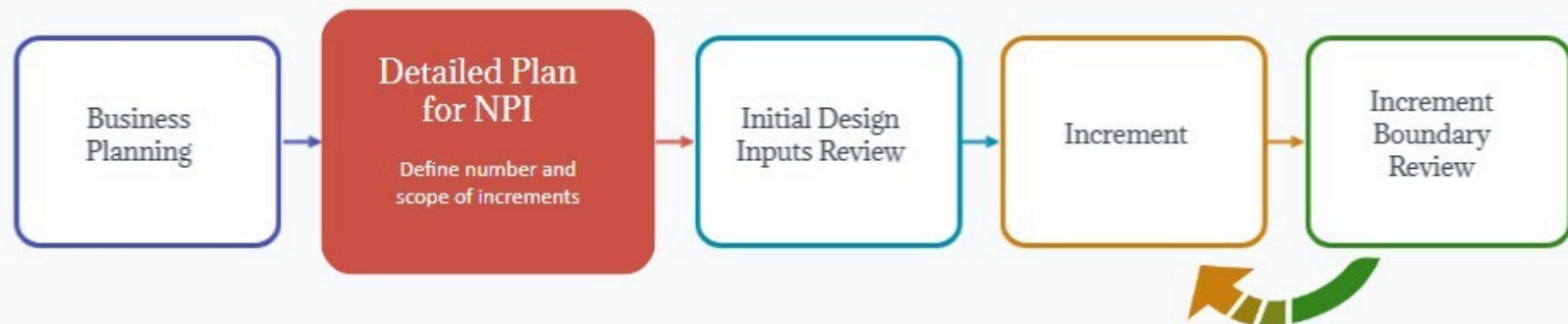
# Agile Design Controls Process



Once the business plan is agreed upon, the R&D Program Manager, supported by various functional teams, develops a detailed plan for software product NPI [7.3].



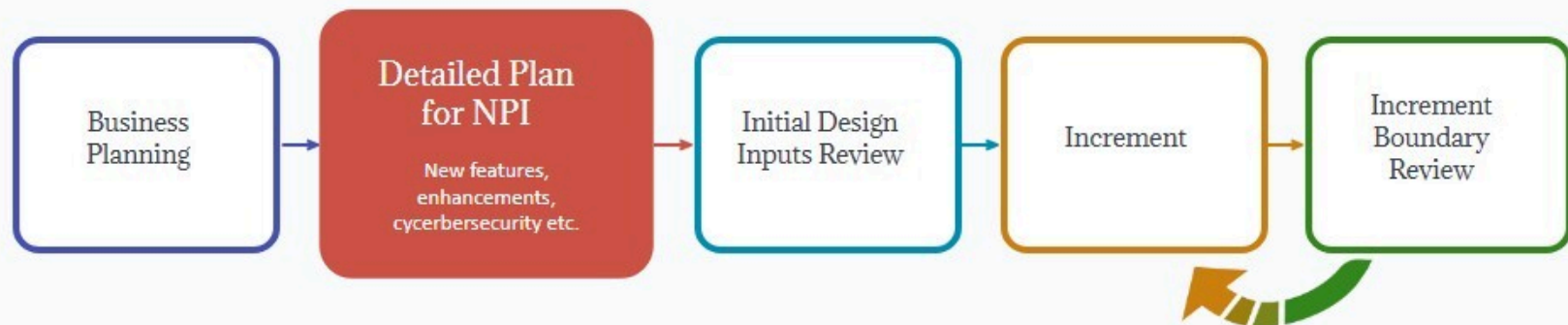
# Agile Design Controls Process



The product plan for a NPI, also known in ADC as the Design and Development plan (D&D plan) may define the number of increments needed to release shippable product, and scope of each increment (eg. features or defects to be fixed).



# Agile Design Controls Process



The development team can decide the level of detail to add in the planning documents from the Agile design controls process 7.3.200G01 and activities as shown in the ADC Agile Development Framework.

CLICK [HERE](#) TO VIEW THE FRAMEWORK.



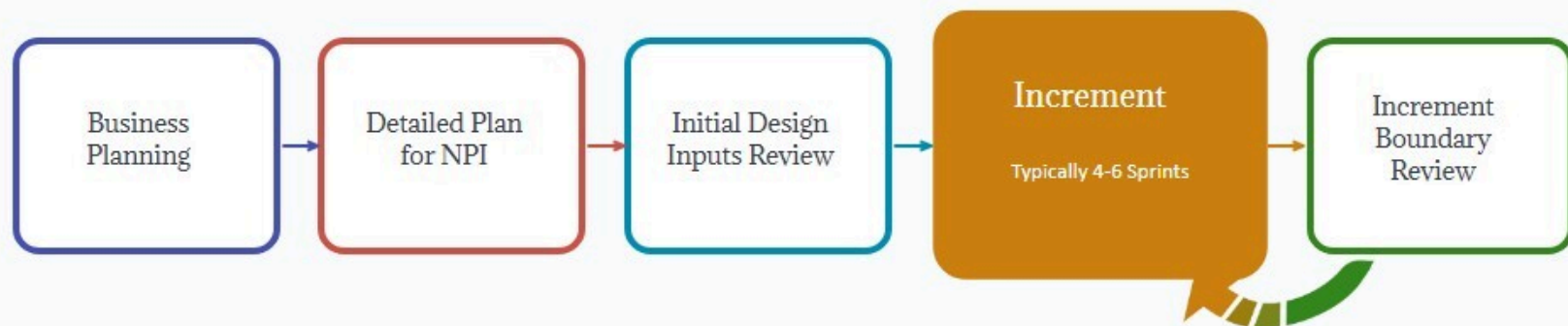


# Agile Design Controls Process



An Initial Detailed Design Inputs review per 7.3 procedure is conducted prior to or early in 1st increment execution.

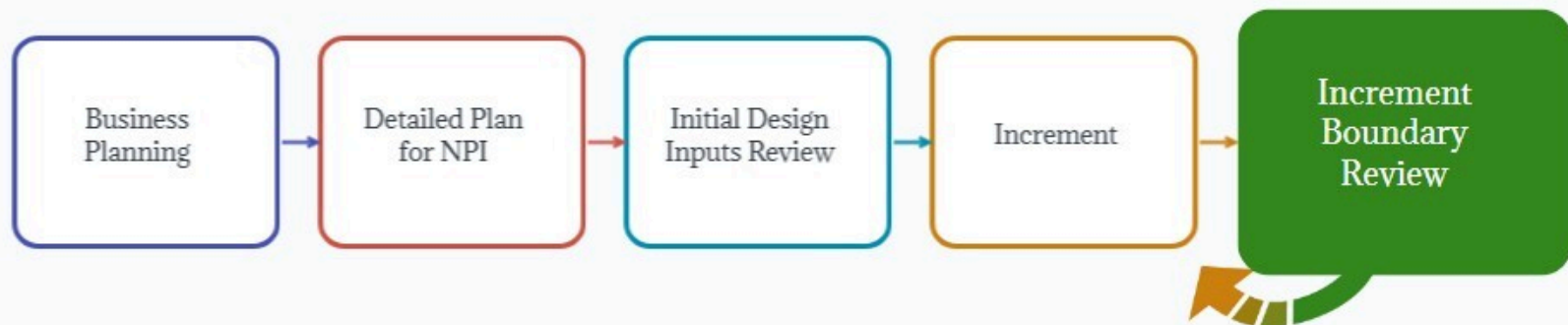
# Agile Design Controls Process



In Sprints, the development team engages in planning, requirements development, risk analysis, architecture design, detailed design, and testing iteratively until the increment or release scope is complete.



# Agile Design Controls Process



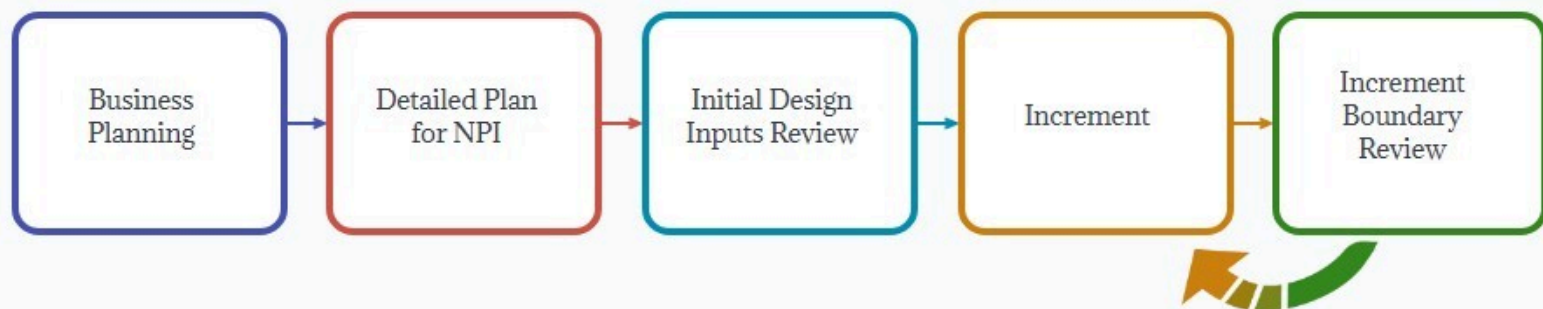
As development progresses, at each increment boundary, additional formal design reviews are conducted per 7.3, including a product demo of functions and features developed.

Development for subsequent increments is modified based on Senior Management feedback in the reviews.



## AGILE DESIGN CONTROLS PROCESS

# Agile Design Controls Process

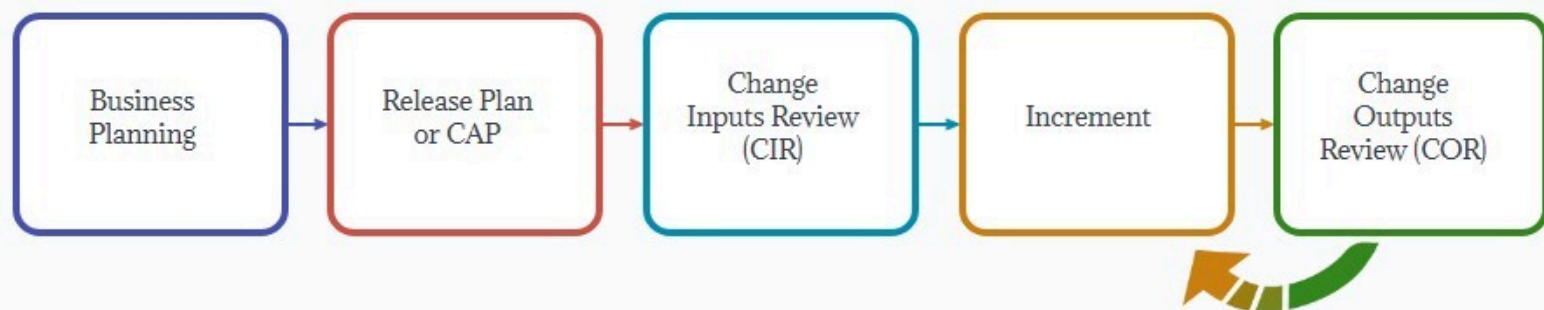


Refer to Agile design controls software guidance 7.3.200G01 for additional process details and deliverables to be generated including systems of record (eg. Agile PLM, DNG, ETM).

The guidance document refers to other ADC design controls procedures, as required.



# Design Changes



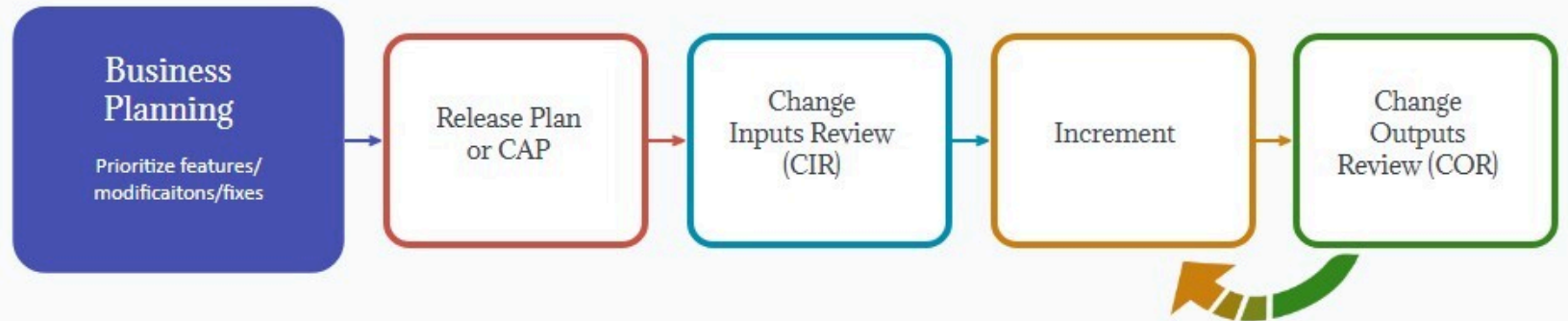
Next, let's take a look at the Agile design controls process for design changes (for standard change in 7.3Wo6).

Note that the process flow we will review over the next few screens represents a high-level overview. For more detailed information on design change process for Agile, please refer to Agile design controls software guidance document 7.3.200G01.





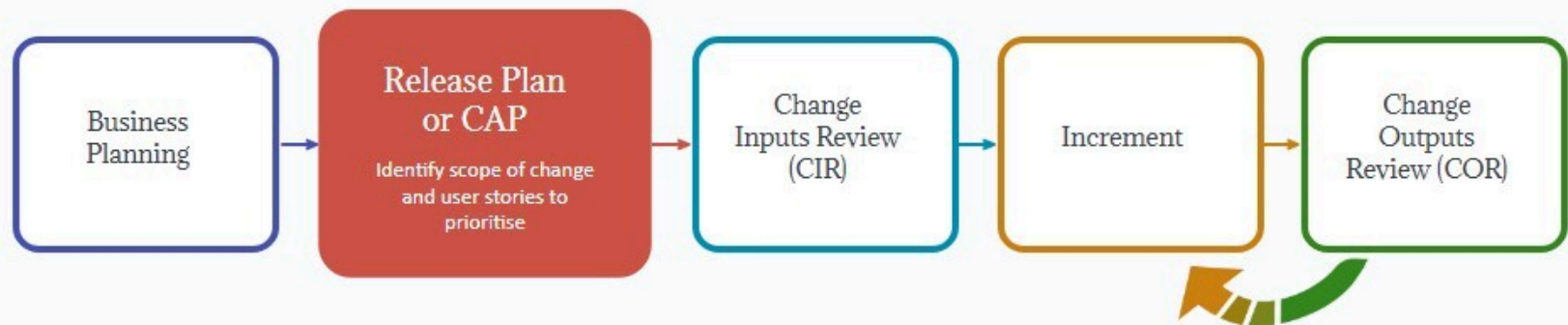
# Design Changes



The R&D Program Manager together with other members of the business planning team meet to prioritize a list of user/customer-meaningful features, modifications or fixes that are planned for the software release.



# Design Changes

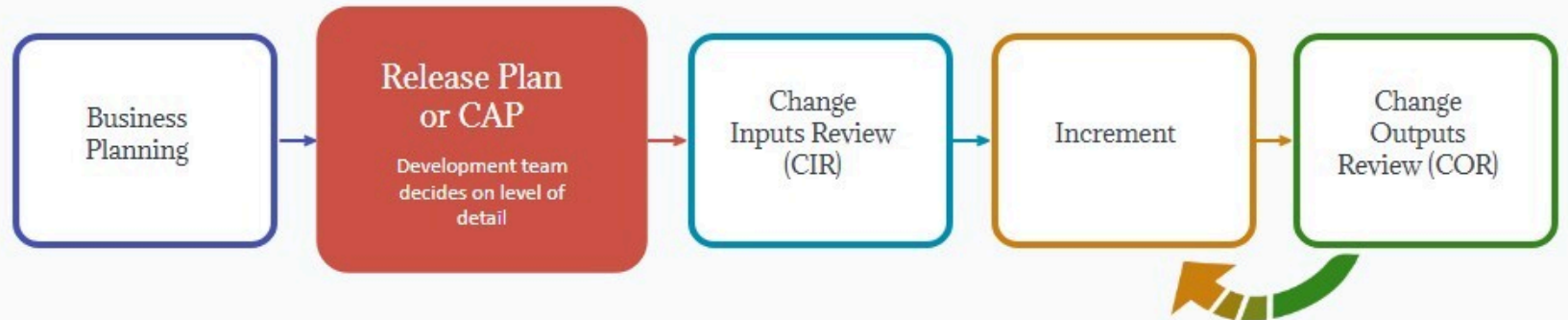


A release plan or a Change Action Plan (CAP) is created to define the scope of the design change based on the prioritized backlog items.

The release plan may also include details on how the increments are organized for the release – that is, the number of increments and the scope of each, DOR/DOD criteria including testing to be performed.



# Design Changes



The development team may decide the level of detail to be added in the planning documents from the Agile design controls process 7.3.200Go1.

# Design Changes

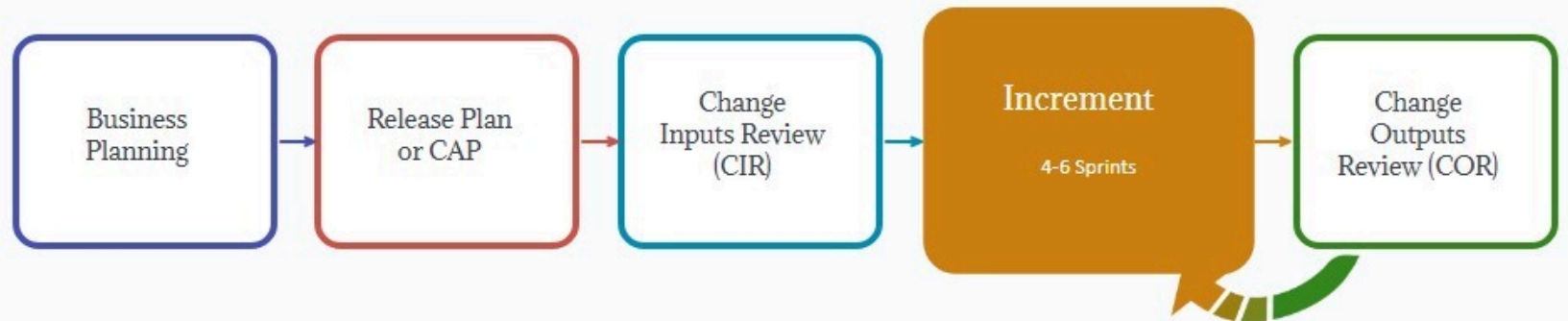


A Change inputs review (CIR) is conducted to present the scope of the change, the change action plan and modified design inputs per 7.3Wo6.

For agile development, the CIR is conducted prior to or early in the first increment execution.



# Design Changes



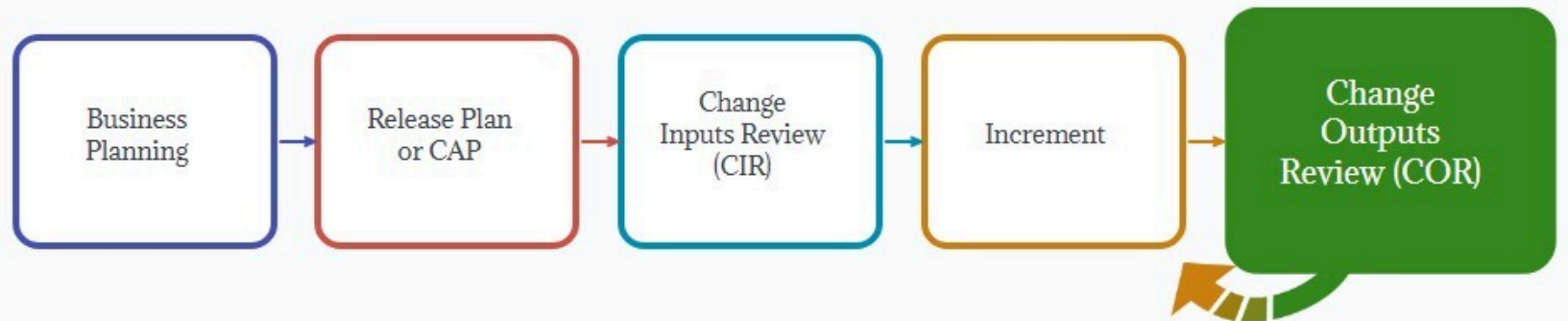
There are typically 4 to 6 Sprints per Increment.

In Sprints, the development team engages in planning, requirements development, risk analysis, architecture design, detailed design, and testing iteratively until the increment or release scope is complete.





# Design Changes

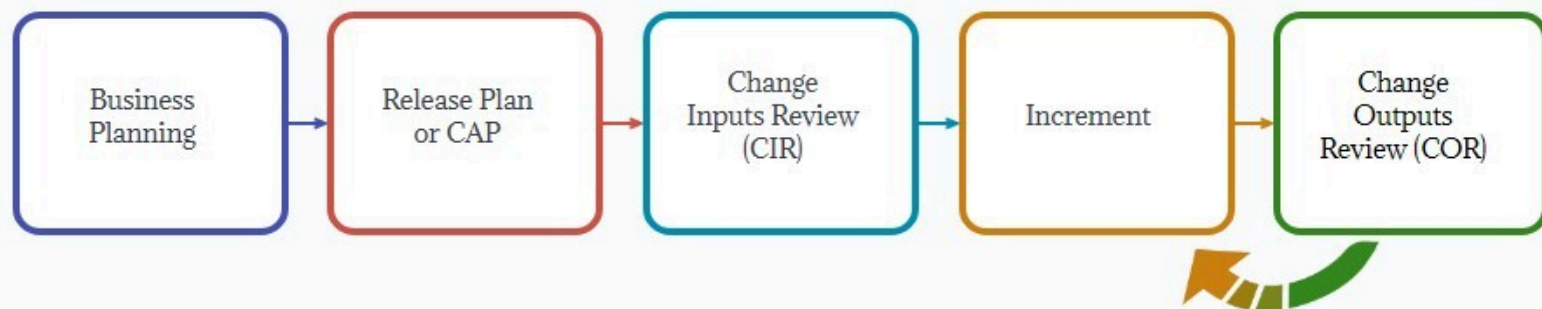


A Change Outputs Review (COR) per 7.3Wo6 is conducted before release.

A product demo is also presented to leadership.



# Design Changes



Refer to Agile design controls process guidance 7.3.200Go1 for additional process details for design changes for Agile software development.





# Review



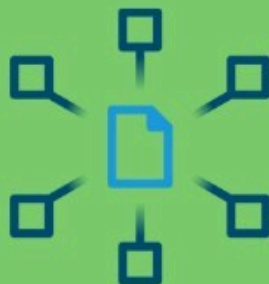
## Review

Take a moment to review some of the key concepts in this section.

Click the arrow to begin your review.



# Review



For NPIs and design changes, the development team develops a detailed plan and engages iteratively throughout the development process.





# Review



Always refer to Agile design controls software guidance 7.3.200G01 for process details.







AGILE

Re

To check your progress, click  
the Menu button



You have completed section 2 of 3

CLICK THE FORWARD ARROW TO CONTINUE LEARNING



## KNOWLEDGE CHECK

# Introduction

---



The Knowledge Check that follows consists of 5 questions. You must score 80% or higher to successfully complete this course.

WHEN YOU ARE READY, CLICK THE KNOWLEDGE CHECK BUTTON.

**KNOWLEDGE CHECK**





## KNOWLEDGE CHECK

# Assessment



## 1

Agile software development follows which lifecycle?

1 | Waterfall

2 | Incremental

3 | Incremental and evolutionary

NEXT

1

2

3

4

5





## KNOWLEDGE CHECK

# Assessment



## 2

\_\_\_\_\_ is the compilation of all activities necessary to begin work on a story, increment and release.

1 | Definition of Done

2 | Definition of Ready

3 | Definition of Feature

NEXT

1

2

3

4

5



## KNOWLEDGE CHECK

# Assessment



## 3

Activities and deliverables for Agile software development is defined in

- 1 | Software design controls procedure, 7.3.200
- 2 | Development lifecycles and frameworks guidance, 7.3G01
- 3 | Agile software design controls guidance, 7.3.200G01

[NEXT](#)

1

2

3

4

5



## KNOWLEDGE CHECK

# Assessment



## 4

A user STORY is a short, simple description of a feature told from the perspective of

1 | the person who desires the new capability

2 | the software developer

3 | the product owner

NEXT

1

2

3

4

5







## KNOWLEDGE CHECK

# Assessment



## 5

5. ADC's Agile design controls process is well aligned to

- 1 | FDA guidance, content of pre-market submissions for device software functions
- 2 | FDA guidance, AAMI TIR45: 2023, guidance on the use of Agile practices in the development of medical device software
- 2 | FDA guidance, general principles of software development

SUBMIT

1

2

3

4

5





## Where to Get Help



### MANAGER OR SUPERVISOR

If you have questions or concerns about an activity or interaction, the best place to start is with your manager or supervisor.

## Course Resources



### TRANSCRIPT

Click [here](#) for a full transcript of the course

2 | FDA guidance, general principles of software development

SUBMIT

1

2

3

4

5

