

# בודק תוכנה Agile - להסמכת ISTQB

קורס מספר 5466 – 32 שעות

## אודות הקורס

קורס זה מבוסס על שיטת הפיתוח המקובלת בימנו ומאפשר שיפור מיומנויות הבדיקה. בקורס זה נלמד כיצד שיטת בדיקה Agile משפרת את מהירות המסירה של הגרסאות בשוטף ומשפרת את האיכות הכוללת של הפרויקט. לימוד וחשיפת המשתתפים ליעדי עולם ה-Agile ובכך ישפרו את יכולות המעבר, והיישום בבדיקות בסביבת Agile ואת הידע הנדרש בפרויקט פיתוח של Agile. בקורס זה נתמקד כיצד מפתחים מוצר תוכנה בעולם האג'ילי ובתפקידים של הצוותי העבודה השונים על פי מתודולוגיית אג'יל.

אנו מוצאים עצמנו בעולם ה-Agile – מתמודדים עם כמויות שינויים נדרשת רבה מאוד ורכיבי התוכנה מפותחים באופן מצטבר בכדי לקדם ולא לייצר עיכובים במסגרת העבודה בעולם אג'ילי אנו נחשוף ונבחן את דרכי היישום הקיימות כגון SCRUM, אקסטרים ו-DevOps, חיזוק התקשורת ומערכות היחסים בין אנשי הצוות על מנת לחזק את התחושה של מטרה משותפת.

בדיקות במודל הפיתוח ה-Agile אינן חלק נפרד במכלול הפרויקט אלא חלק משולב של צוות הפיתוח. כיצד מבצעים בדיקות במקביל לעבודת המפתחים, מה משמעות צוותי פיצ'ר, והשפעתם על בדיקות אינטגרציה בשלבים מוקדמים מה שמאפשר לספק ללקוחות כמות של רכיבי תוכנה מאושרים בצורה הדרגתית. עולם הפיתוח ה-Agile פתח צוהר משמעותי לדרכי פיתוח שונים והזדמנויות לבניית מוצר איכותי, תוך התמקדות בערך לקוח, שיפור תהליך הבדיקות, שיפור תהליך העבודה, והאמון בין חברי הצוות.

כאשר אנו מבצעים שינוי בשיטת / גישת פיתוח אנו עדיין עלולים ל"שמר" הרגלים ישנים ולא באמת לבצע מעבר בפועל. אנו יכולים למצוא עצמנו מצהרים על מעבר תפישתי ללא ביצוע מחקר והערכה והתחייבות למה שבאמת עלינו לשנות. בפועל ניתן לממש את רב הבדיקות רק במסגרת השלבים מתאימים, מה שעומד בניגוד לשיטת הבדיקות ב-Agile. נלמד ונתמקד בשינויים שיש לבצע בכדי ליישם את עקרונות הבדיקות בצורה הטובה ביותר.

הקורס מכין בצורה מלאה למבחן ההסמכה היוקרתית של ISTQB המעניק למסיימים בהצלחה תעודה רשמית בינלאומית.

## דרישות קדם

- היכרות בסיסית עם עולם פיתוח התוכנה, או מינימום חצי שנת ניסיון בתחום בדיקות תוכנה או פיתוח תוכנה.
- ניסיון עבודה באחד התחומים הבאים: Agile Testers, מפתחי Agile, מנהלי מוצר, אנליסטים ומנהלי בדיקות, מנהלי QA ואלה שתפקידם להבטיח כי צוותי Agile יספקו שביעות רצון גבוהה של לקוחות בעלות אופטימלית.
- אנגלית אקדמית בסיסית
- על מנת לגשת לבחינת ההסמכה Agile של ISTQB, חובה ראשית לעבור את ההסמכה ISTQB Foundation Level.

## קהל יעד

- מנהלי פרויקטים, מנתחי מערכות, אנליסטים עסקיים
- מנהלי פיתוח תוכנה, מתכנתים / מפתחים.
- אנשי QA, אנליסטים ומהנדסי איכות, בודקי תוכנה, מנהלי איכות תוכנה ובדיקות וכל גורם המעורב בתהליכי הפיתוח והבדיקות.

## מטרות הקורס

- להבין את ההבדלים מרכזיים בין שיטות בדיקה מסורתיות ו Agile.
- שינוי התפישה ויצירת מיקוד חזק בשותפות וקשרי לקוח.
- מיקוד בהבנת הדרישות והפיכתם למקרי בדיקה איכותיים
- הכרת האתגרים התרבותיים והלך הרוח של עבודה מוצלחת ב Agile (פיתוח וצמיחה של צוותי פיצ'ר המנהלים את עצמם)
- כיצד לבנות ולפתח תכנית מעבר משיטות הבדיקה המסורתית לבדיקות Agile.
- היכרות השווה והשונה בין גישות Agile הפופולריים (כולל Scrum ו Kanban)
- יישום והטמעת הערכים והעקרונות של מודל Agile לפיתוח מוצרים במסגרת בדיקות התוכנה
- שיטות לבניית מסגרת בדיקה רזה (Lean) כדי לשפר הגשת גרסאות Agile.
- כיצד להתנהל במסגרת אילוצי הזמן מבלי לאבד זמן בדיקה יקר.
- שימוש בשיטות בדיקה מהירות בכדי להוסיף ולתמוך בשיטות בדיקה המבוססות על סיפורים (Exploratory, Pairwise, Pareto)
- לייעל את פיתוח הבדיקות באמצעות שימוש וניהול שימוש חוזר.
- כיצד משלבים את צוות הבדיקות בפרויקט Agile
- בניית בדיקות מבוססות אובייקטים עבור בדיקות רגרסיה
- לסייע למנהלים בקבלת החלטות איכותיות על בסיס תוצרי הבדיקות
- להיחשף לאפשרויות התמיכה האוטומטית.
- הכנת המשתתפים לבחינת ההסמכה הבינלאומית ISTQB ברמת הבסיסית ולימוד כל הרקע המתודולוגי הנדרש לאנשי QA.

## תכני הקורס

### 1. Agile Software Development

#### 1.1 The Fundamental of Agile Software Development

##### 1.1.1 Agile Software Development and Agile Manifesto

##### 1.1.2 Whole Team Approach

##### 1.1.3 Early and Frequent Feedback

##### 1.1.4 Typical Programmer Role in Testing

#### 1.2 Aspects of Agile Approaches

##### 1.2.1 Agile Software Development Approach

##### 1.2.2 Collaborative User Story Creation

##### 1.2.3 Retrospective

##### 1.2.4 Continues Integration

### 1.2.5 Release and Iteration Planning

## 1.3 Agile Testing Mindset

### 1.3.1 Overview of Agile Testing

(Origins of Agile Testing, Agile Testing vs. Traditional Approaches)

### 1.3.2 Mindset and Culture

(Agile Testing Principles, Whole Team Approach, Building Quality In, Continuous Improvement and Feedback, Ingraining the Agile Testing Mindset)

## **2. Fundamental Agile Testing Principles, Practices and Processes**

### 2.1 The Differences between Testing in Traditional and Agile Approaches

#### 2.1.1 Testing and Development activities in Agile projects and non-Agile projects

#### 2.1.2 Project Work Products

#### 2.1.3 Test Level

#### 2.1.4 Testing and Configuration Management

#### 2.1.5 Organizational Option for Independent Testing

### 2.2 Status of Testing in Agile Projects

#### 2.2.1 Communicating Test Status, Progress, and Product Quality

#### 2.2.2 Managing Regression Risk with Evolving Manual and Automated Test Cases

#### 2.2.3 Defect Tracking and Management

#### 2.2.4 Results Reporting

#### 2.2.5 Test Metrics Successful Delivery

#### 2.2.6 Time-Boxed Delivery

#### 2.2.7 Continuous Delivery

#### 2.2.8 Post-Development Test Cycles

#### 2.2.9 Iteration Wrap-Up

#### 2.2.10 Definition of a Release/End Game

#### 2.2.11 System-Wide and Cross-Team Testing

#### 2.2.12 Post-Release Testing

#### 2.2.13 Documentation for Regulatory Requirements

#### 2.2.14 Test Environments and Infrastructure

### 2.3 Role and Skills of a Tester in Agile Team

#### 2.3.1 Agile tester Skills

#### 2.3.2 The Role of a Tester in Agile Team

### 3. Agile Testing Methods, Techniques, and Tools

#### 3.1 Agile testing methods & Strategy

3.1.1 Test Driven Development, Acceptance Test Driven Development, and Behavior Driven Development.

3.1.2 The Test Pyramid

3.1.3 Testing Quadrants, Test Levels, and Testing Types

3.1.4 Test Strategy and Planning

3.1.5 Different Strategies Based on Levels of Precision

3.1.6 During Iteration Planning/Kickoff

3.1.7 Lightweight Test Plan Documentation

3.1.8 The Role Of Tester

#### 3.2 Assessing Quality Risks and Estimating Test Effort

3.2.1 Assessing Quality Risks in Agile Projects

3.2.2 Estimating Testing Effort Based on Content and Risk.

#### 3.3 Techniques in Agile Projects

3.3.1 Acceptance Criteria, Adequate Coverage, and Other Information for Testing

3.3.2 Applying Acceptance Test-Driven Development

3.3.3 Functional and Non-Functional Black Box Test Design

3.3.4 Exploratory Testing and Agile Testing

#### 3.4 Feature and Story Testing

3.4.1 User Story Testing

3.4.2 Feature Testing

3.4.3 Exploratory Testing

3.4.4 Non-Functional Testing

#### 3.5 Testing Environment

3.5.1 Typical Environments for Test

3.5.2 Build Pipeline

3.5.3 Automated Builds

3.5.4 Testing the Proper Build

3.5.5 Test Data Management

3.5.6 Working on Distributed Teams

3.5.7 Distributed Team Communication

3.5.8 Distributed Team Coordination

### 3.6 Tools in Agile Projects

3.6.1 Task Management and Tracking Tools

3.6.2 Communication and Information Sharing Tools

3.6.3 Software Build and Distribution Tools

3.6.4 Configuration Management Tools

3.6.5 Test Design, Implementation, and Execution Tools

3.6.6 Cloud Computing and Virtualization Tools