

קורס פיתוח ובדיקות Agile

תאור הקורס

מסלול הכשרה ייחודי זה, מקנה ידע נרחב בפיתוח, בדיקות תוכנה וניהול Agile, ומכין את התלמיד לעסוק בתחום מבוקש זה, באופן מהיר וממוקד לצרכי השוק בתחום. ביישום של תוכנה טובה יותר מעורבים שלוש מטרות: המוצר הנכון, נעשה נכון, נוהל נכון. בקורס נלמד כיצד לספק את כל השלושה. כדי לספק תוצאות ב-agile, צריך הרבה יותר מללמוד את העקרונות וליישם את תהליך agile; **צריך ליצור תרבות של התנסויות**. המחויבות הזאת להתנסות נמצאת במרכז הקורס שמטרתו ללמד כיצד להתנהל ברמה גבוהה בפרויקטים של agile. קורס זה מקנה לתלמיד את היכולת לשלב, להתמודד עם השינויים ולהבין את התפיסה של השינוי המהיר בבדיקות פיצ'רים ורכיבים בדידים, בבדיקות שמישות ובדיקות התוכנה עצמה.

התלמיד ילמד לענות על ארבע שאלות מרכזיות בנוגע לתוצר:

1. האם צריך לבנות אותו?
2. האם התוצר משנה משהו?
3. האם התוצר שמיש?
4. האם הוא תקול?

לאחר השלמת הקורס התלמיד יוכל:

- ✓ לזהות היכן וכיצד להשקיע את הזמן והאנרגיה בביצוע בדיקה טובה יותר להשפעה מקסימלית על התוצאות
- ✓ להבין את הקשר בין דרישות ובדיקות כדי ליצור את שיתוף הפעולה הנדרש בין הבודקים ללקוחות
- ✓ להבין את הקשר בין שימושיות ובדיקות תוכנה כדי ליצור את שיתוף הפעולה הנדרש בין הבודקים ליתר חברי הצוות
- ✓ לבדוק רעיונות לפני שבונים את תהליך הבדיקות או הפיתוח, כדי למנוע בזבז משאבים ולעזור לצוות להתמקד במה שבאמת נחוץ להשגת התוצאות הנדרשות
- ✓ לבחון פתרונות וממשקים חלופיים טרם בניית המוצר, על מנת למקסם את השימושיות של המוצר הנדרש
- ✓ להבין את תהליך אספקת המוצרים וכיצד לתעדף שיפורים בתהליך, כך שאפשר לשפר את תהליך אספקת מוצר לענות על הצורך באספקה מהירה, איכותית ולעתים קרובות יותר

קהל יעד

- כל המעוניינים לרכוש מקצוע נדרש ולעבוד בתחום ההייטק
- בעלי תעודת בגרות וכל המעוניינים לעסוק בתחום בדיקות התוכנה פיתוח תוכנה או ניהול בתחום ה-Agile

דרישות קדם

- ידע וניסיון במערכות הפעלה חלונאיות ובתפעול מחשב.
- שליטה טובה באנגלית (חלק מחומר הקורס הינו באנגלית).
- פגישת ייעוץ ובדיקת התאמה לתחום

דרישות הקורס

- נוכחות ב 80% מהמפגשים לפחות
- הגשת תרגילי בית ופרויקטים
- מעבר בחינות פנימיות

משך הקורס

48 שעות אקדמיות

תעודות

בסיום הקורס ולאחר עמידה בכל דרישות הקורס יקבלו הסטודנטים תעודת גמר של המרכז הישראלי לטכנולוגיות הבטחת איכות IITC-QA

תוכנית הקורס

1. הכול מתחיל ב lean (8 שעות אקדמיות)

הבסיס של פיתוח זריז נעוץ בתפיסת העולם של לין lean - ניהול ותפעול רזה. בפרק זה נביר את המושגים- ערך ובזבוז. עמוד השדרה של agile הוא שאפשר לצפות את התוצאות, וזאת בניגוד לחוסר הוודאות הקיימת ביצירת תוכניות מורכבות של פרויקטים המבוססים על מודלי פיתוח אחרים. התלמידים לומדים את משמעות הבזבוז והדרך להימנע ממנו, על אימות בעיות / צרכים ומוטיבציה של המשתמש לפני השקעה בתוכנה. הבדיקות בפועל מחייבות מיומנות וסוג שונה של הבנת תפיסת העולם של הבדיקות המאפשר לחלק את העבודה לחלקים קצרים וברי לבדיקה. במודול זה התלמיד יבין מהי דרך היישום של תפיסת החיים הרזה ומשמעותה בביצוע בדיקות. העמקת בטכניקות התפעוליות של לין ואת המוטיבציה של בדיקות בסביבה רזה. התלמיד ילמד כיצד איך לעצב ולהפעיל בדיקות מותאמות מצב, וכיצד זה עובד: בבדיקת הרעיון, ובבדיקת הערך של התוכנה העובדת.

- מהו לין & והבדיקות
- יצירת תרבות של בדיקות רזה
- על הפרקטיקה של ה- Lean
- העבודה עם Lean

- הפרקטיקה של lean
- יצירת תרבות מבוססת בדיקות ומשמעותה
- בדיקות מונחות עיצוב
- מעקב אחר השערות
- שילוב מתודולוגיית Agile עם תפיסת העולם הרזה
- עיצוב הבדיקות
- תכונות בדיקה: מתי וכיצד, הפעלת הבדיקות, מעקב, ותוצאות הבדיקה
- תרגול – האם צריך לבנות זאת, האם זה משנה?

2. ניהול דרישות - 8 שעות אקדמיות

כדי לזהות את המוצר הנכון, צריך להבין את הצרכים של הלקוחות ושל המשתמשים ולא רק את מה שהם רוצים.

בשיעור זה נתמקד מהי בעיה שצרכים לפתור? מה הן המשימות שיש לבצע כדי להבין את הדרישות. מי ישתמש בתוכנה זו וכיצד תהיה אינטראקציה עם המשתמש? מענה על שאלות אלה, יאפשר לצוות לבנות את המוצר הנכון.

כדי להשיג את לבצע נכון ולנהל נכון, פיתוח תוכנה חייב להתחיל בדרישות בהירות וברורות לפיתוח התוכנה, אשר בשלבים יותר מאוחרים, יבוצע התכנון, היישום והבדיקות על פיהם.

בשיעור זה נתמקד כיצד להבין את הצרכים מהלקוחות ומהמשתמשים. כמו כן תלמד כיצד לבטא צרכים אלה כמערכת של דרישות המאפשרת פיתוח תוכנה, פיתוח ועיצוב בדיקות ותכנון הפעילות הנדרשת.

• התמודדות עם דרישות משתנות

כיצד מתמודדים עם דרישות משתנות, וכיצד להימנע לשינויים בלוחות הזמנים ליישום הדרישות scope creep.

נתמקד בהבדלים שבין דרישות ועיצוב תהליך הבדיקות או הפיתוח

• הבנת הדרישות

נתמקד בבעיות שבהבנת דרישות המשתמשים ונלמד דרים יעילות להבנת הצרכים, ומה צריך המוצר לעשות. מהן דרישות טובות והמה הבעיה עם דרישות עמומות

• טכניקות

היכרות עם טכניקות – wire frames ו- storyboards .

היכרות עם ניהול דרישות זריזות Agile requirements techniques

- User stories - סיפורי משתמשים לתיאור הדרישות
- בדיקות קבלה acceptance tests – לאישור הדרישות
- ניהול ותעדוף הדרישות backlog
- מפות סיפור story maps לארגון הבדיקות

עבודה – ניהול דרישות

3. תכנון פרויקט זריז - 8 שעות אקדמיות

מהן האסטרטגיות והטכניקות לתכנון פרויקט בדיקות התוכנה שלכם. טכניקות אלה יהפכו את הפרויקט שלך למותאמים לשינויים מכיוון שהם מיושרים עם מניפסט זריז.

a. חלוקת המשימות למשימות ניהול קטנות

המפתח לתכנון פרויקט תוכנה זריז הוא לשבור את עומס העבודה למשימות קטנות יותר לניהול. בשיעור זה נתמקד כיצד לתאר ולארגן משימות ברמה פרטנית ולהתמקד כיצד לפרק פעילויות גדולות יותר. כיצד ליצור הערכות עבודה ראליות, ולתזמן באופן מדויק את כמות העבודה בספרינט.

b. טכניקות

- היכרות עם טכניקות ניהול פרויקטים, Release Plans, Charts, Gantt
- היכרות עם טכניקות ניהול איטרציות - PERT Charts, CPM Charts and Iteration Plans
- טכניקות ניהול על פי ניהול סיכונים

4. בקורות ומדדים - 8 שעות אקדמיות

בשיעור זה נבחן מהם המדדים והבקורות שיש לבצע כדי להגדיר ולספק את המוצר הנכון, שנעשה נכון, והצליח נכון.

נלמד אלמנטים לקביעת המוצר הנכון, וכיצד להבטיח שאתה מספק את המוצר הנכון.

- לימוד על פגישות סקירה בספרינט. סוג זה של ביקורת מאמתת שיוצרים את המוצר הנכון עבור הלקוח.
- נסקור את מחקרי המשתמשים user studies. מחקרי משתמשים הם דרך מצוינת לאמת שאנחנו יוצרים את המוצר הנכון עבור המשתמשים במוצר.
- החשיבות של אסטרטגיית המוצר, מחזור החיים וניהול שיווק בניהול מוצר

5. מהי שימושיות וכיצד בודקים אותה - 8 שעות אקדמיות

a. טכניקות שיסייעו בעבודה השוטפת

לימוד טכניקות שיסייעו בעבודה השוטפת:

- בדיקות מונחות עיצוב מוצר
- בדיקות מונחות עיצוב מוצר ותמיכה של המערכת בדרישות משתמשים
- בדיקת שימושיות
- כיצד יוצרים Test Suite
- כיצד יוצרים Test Items
- מהם סיפורי משתמשים Agile User Stories
- הרצת בדיקות שימושיות
- יצירת אב טיפוס אינטראקטיבי

- מחקר איכות לעומת כמות
- כיצד מתקנים שימושיות?
- יצירת עקביות
- מונחים נפוצים: Wireframes, Mockups, אב טיפוס Prototypes, תרגול: האם זה שמיש?

6. בדיקות כטבע שני - 8 שעות אקדמיות

- בדיקות מונחות פיתוח תוכנה
- ניהול גרסאות
 - ההתחייבות ובדיקות באיטרקציות קטנות מועד
 - בקרת גרסאות
 - ניהול מונחה בדיקות
- תהליך פיתוח ואספקה רציף
 - תחילת העבודה תהליך פיתוח ואספקה רציף
 - היתרונות של תהליך פיתוח ואספקה רציף
 - עיצוב מונחה יישום רציף
 - כיצד יוצרים סביבת עבודה ידידותית התומכת בתהליך פיתוח ואספקה רציף?
- בדיקות של פרויקטים שונים
 - בדיקות של פרויקטים בינוניים
 - בדיקות של פרויקטים גדולים
 - יצירת בדיקות התנהגות לפרויקטים גדולים
- בדיקות יחידה Unit Testing
- בדיקות פונקציונליות
- בדיקות ממשק משתמש
- יישום והטמעה deployment
- שלב השחרור - release
- הופעת ה- Devops והעלייה של יישום רציף Continuous Delivery