

SDET\QA Architect 2020

מסלול בדיקות תוכנה ידניות ואוטומציה + ארכיטקטורת תוכנה

תיאור המסלול

מסלול הכשרת בודקי תוכנה במכללת INT הינו מסלול מקיף ועדכני המכשיר את המשתתפים לידע נרחב בתחומי בדיקות תוכנה בדרך אוטומטית או ידנית. המסלול מתאים גם לחסרי רקע טכנולוגי המעוניינים להשתלב כבודקי תוכנה בארגונים ולפתח קריירה בתחום זה. מסלול הלימודים כולל לימודים עיוניים ומעשיים המקיפים את עולמות התוכן בבדיקות התוכנה הן בהיבט המתודולוגי והן בהיבט המעשי יישומי. במסלול נכללים תרגולים, בחינות, סימולציות ופרויקטים המכשירים את הסטודנטים להתמודדות עם מצבי אמת. מכללת INT היא המכללה המובילה את תחום הכשרת בודקי התוכנה בישראל ובמהלך השנים הוכשרו במכללה אלפי סטודנטים שהשתלבו בתעשייה ופיתחו קריירה מצליחה כמובילי תהליכי הבטחת איכות בארגונים.

מסלול זה מחולק לשלושה שלבים:

שלב ראשון - מבוא לעולם בדיקות התוכנה ועולם הבדיקות הידניות - 155 שעות אקדמאיות (140 שעות פרונטליות, 15 שעות תרגיל להגשה בעבודה עצמית).

שלב שני - עולם בדיקות האוטומציה והכנה למבחני הסמכה ISTQB - 255 שעות אקדמאיות (175 שעות פרונטליות, 80 שעות תרגילים ופרויקט גמר להגשה בעבודה עצמית).

שלב שלישי - עולם ארכיטקטורת התוכנה - 120 שעות אקדמיות (110 שעות פרונטליות, 10 שעות פרויקט מסכם בעבודה עצמית).

חשוב לדעת

- **היקף השעות:** 530 ש"א (425 שעות לימוד פרונטליות, 105 שעות לימוד תרגול עצמי)
- **קהל יעד ודרישות קבלה:** המסלול מיועד לחסרי רקע בעלי ניסיון עבודה עם מחשבים ברמת משתמש, הבנה בסיסית ביישומי Office ושליטה בשפה האנגלית המבקשים להשתלב בתחום בדיקות האוטומציה או בדיקות ידניות.

- **מתודולוגיית הלמידה:** המסלול מורכב ממספר נושאים. לכל מודול יש חלק תאורטי וחלק מעשי. השיעור יורכב מהסבר תאורטי וממקבץ תרגילים מעשיים אשר יבטיחו את הבנה והטמעה בטוחה ויציבה של החומרים. בסוף כל פרק, תתבצע בחינה לסיכום החומר הנלמד. כמו כן, ישנו פרויקט מסכם הכולל את כלל החומר הנלמד במהלך המסלול. ההצלחה שלכם במסלול היא ההצלחה שלנו.
- **סגל המרצים:** למכללת iNT סגל מרצים ומומחי הדרכה, מהמובילים בתחומם, בעלי ניסיון מעשי רב ביישום והדרכת נושאי הלימוד בתעשיית ההי-טק הישראלית והעולמית.
- **זכאות לתעודת גמר מטעם מכללת iNT:** קבלת תעודת בוגר מסלול מטעם המכללה מותנת בהשתתפות בלפחות 85% מהשיעורים ומעבר כל המבחנים בציון עובר בנוסף לעמידה בתקנון הלימודים.

**שלב א' - מבאות
ובדיקות ידניות 155 ש"א**

מודול 1 - מבוא למחזור חיי בדיקת תוכנה (20 ש"א)

במודול זה נלמד את מושגי הבדיקה הבסיסיים. בנוסף, נבין את מחזור החיים של פיתוח תוכנה (SDLC - Software Development Life Cycle), נכיר את עולם הבדיקות ואת תפקיד הבודק. נבין את הקשר בין בודקי התוכנה אל מול המפתחים וכיצד מתבצע החיבור בין עבודתם. בנוסף, נבין את עקרונות הבדיקה כמו: מהם מסמכי הגדרת בדיקות, מי עורך את מסמכי הגדרת הבדיקות, מי יוזם את הבדיקה, מה הטריגרים לבדיקה ומי מבצע את הבדיקה, מי הם מקבלי ההחלטות ומה הם הצעדים הנדרשים לאחר קבלת תוצאות הבדיקה.

מודול 2 - תכנון בדיקה (15 ש"א)

במודול זה, נלמד כיצד לנהל את הבדיקה, נלמד איך להחליט אילו מחלקי התוכנה מקבלים עדיפות גבוהה יותר לפיתוח בדיקות עבורם (מבחינת ניהול סיכונים). נבין גם את התפקידים והאחריות שממלאים מפתחי הבדיקות ואת המעורבות / התרומה שלהם. כמו כן, נלמד תהליכי בדיקות ומסמכי בדיקות (STP, STD, STR).

מודול 3 - עיצוב בדיקה (10 ש"א)

במודול זה, נלמד ליצור תרחישי בדיקה, לפתח מקרי בדיקה ולזהות נתוני בדיקה מתאימים. בסוף מודול זה, נדע לכתוב תרחישי בדיקה עבור אפליקציה לדוגמה. כמו כן, נלמד להשוות בין התוצאות המצופות מהבדיקה לבין התוצאה שקיבלו בפועל (שימוש ב-Asserts).

תרגיל : כתיבת Test Cases ובחירת נתוני בדיקה עבורם.

מודול 4 - טכניקות בדיקה (10 ש"א)

במודול זה, נלמד את החשיבות של הבדיקה, מדוע לבדוק ואיך, כתיבת מקרי בדיקה יעילים. מודול זה, מעניק לימוד מעמיק להבנת טכניקות כתיבה דינמיות לבדיקות דינמיות לכיסוי מקסימלי עם מינימום מקרי בדיקה.

מודול 5 - רמות וסוגי בדיקות (10 ש"א)

במודול זה, נבין את השלבים והסוגים השונים של הבדיקה. כמו כן, נלמד את ההבדלים בין סוגי הבדיקות השונות והשילובים ביניהן, ומתי נכון להשתמש בכל אחת מהן.

מודול 6 - ביצוע בדיקות (15 ש"א)

במודול זה נבין את קריטריוני הכניסה והיציאה לביצוע בדיקות וכיצד לבצע את מקרי הבדיקה שפותחו.

מודול 7 - ניהול Bugs (15 ש"א)

במודול זה נלמד את ההבדל בין מניעת Bugs וגילוי Bugs. נבין את מחזור החיים של ה-Bug וכיצד לדווח על Bug עם כל הפרטים הדרושים. מונחי מחזור חיי ה-Bug, דיאגרמת מכונת מצבים של Bug, מתי מסתיים הטיפול ב-Bug ועוד.

מודול 8 - דיווח ושיתוף פעולה בצוות (5 ש"א)

במודול זה תלמד להכין דוחות סטטוס על התפתחות מקרי הבדיקה, ביצוע תיקי בדיקה, דוח מצב Bug's, דוחות סגירת בדיקה. בכיתה יוצגו דוגמאות להבנה טובה יותר. נלמד גם כיצד ליצור קשר עם צוותים שונים בארגון ולהפחית אתגרים. נדגים עבודה במערכת לניהול פרויקטים, משימות ותהליכי עבודה כדוגמת JIRA, Test trail וכו'.

מודול 9 - מדדי איכות - KPI (5 ש"א)

בבדיקת תוכנה, חשוב מאוד למדוד את האיכות, העלות והיעילות של הפרויקט בהתאם לבדיקה. במודול זה נלמד מה הם מדדים וכיצד לאסוף מדדים ולמדוד את איכות הבדיקה בהיבטים שונים.

מודול 10 - בדיקות בעולם ה-SAP (15 ש"א)

הכרות עם מערכות SAP ומודולים מובילים ברמת משתמש, כולל ניתוח אפיונים של מערכות SAP וכתיבת תרחישי בדיקה מותאמות למערכות מידע אינטגרטיביות.

פרויקט מס' 1- בסיום מודול זה יבוצע פרויקט בדיקת מערכת מידע הכולל ניתוח מערכת, כתיבת תרחישים והרצתם באחת ממערכות ניהול הבדיקות בהתאם להחלטת המרצה, במתכונת של עבודה עצמית (15 ש"ל)

מודול 11 - בדיקות ידניות בעולם ה-WEB (20 ש"א)

בחלק זה, נבין את עולם ה-WEB. שימושים, יתרונות וחסרונות, פלטפורמות ואתגרים בבדיקות. בנוסף, נלמד מתודולוגיות QA בעולם ה-WEB. מבחן מסכם: בדיקות ידניות (2.5 ש"א)

שלב ב'
בדיקות אוטומציה והכנה
ל-ISTQB 255 ש"א

מודול 1 - מבוא לתכנות (20 ש"א)

מודול זה יציג את עקרונות התכנות ויאפשר לחסרי רקע בתכנות לייצר פתרונות לוגיים ולהבין את הפעולות העיקריות למימוש Flow Control בתוכניות. בשלב זה נכיר בעיות ידועות בעולם התכנות, אלגוריתמים, ונבין את תהליכי פיתוח התוכנה.

מודול 2 - תכנות מונחה עצמים OOP בשפת Java (30 ש"א)

במודול זה נכיר את מתודולוגית התכנות המודרנית המוכרת כתכנות מונחה עצמים. תכנות מונחה עצמים מבוסס על הפרדת מרכיבי התוכנית לישויות עצמאיות המכונות אובייקטים. בחלק זה יילמדו הנושאים המהותיים בתכנות מוכוון עצמים הכוללים הורשה - Inheritance, כימוס - Encapsulation ורב צורתיות - Polymorphism.

מודול 3 - מסדי נתונים - SQL (20 ש"א)

במודול זה ילמדו הסטודנטים עקרונות יסוד בתחום מסדי הנתונים. כגון, בניית מסד נתונים, בנייה ושימוש במודלים של נתונים, ביצוע נורמליזציה לטבלאות, מימוש מודלים של נתונים והמרתם לטבלאות וכן, עבודה עם שפת השאילתות SQL.

MySQL הוא מסד נתונים יחסי, רב נימי ורב משתמשים מבוסס שפת **SQL**. בחלק זה אנו נכיר את מסד הנתונים, סביבת העבודה ונממש את שאילתות ה- **SQL** שלמדנו קודם לכן במסד הנתונים **MySQL**.

מודול 4 - ניהול תצורה והפצת קוד Git (3 ש"א)

Git היא מערכת בקרת גרסאות קוד. בחלק זה, נלמד כיצד לנהל, לאחסן ולהפיץ קוד. נלמד את כלי ה- Git המאפשר ניהול תצורה. נקים ספריה באתר - GitHub ונעלה לשם את הפרוייקטים שנפתח.

מודול 5 - בדיקות אוטומציה בעולם ה- WEB (Selenium & Frameworks) (50 ש"א)

כאשר אנחנו מבצעים בדיקות אוטומטיות חשוב שתהיה לנו שליטה כמה שיותר גדולה על סביבת העבודה שלנו. תשתית Selenium מגיעה על מנת לתת פתרון פשוט, נוח וקל לתפעול ושליטה על דפדפנים. באמצעות השליטה הנ"ל, ניתן יהיו לבצע תרחישי בדיקה אוטומטיים אשר יבטיחו את איכות המוצר אותו אנו בודקים.

תחילה נכיר את מתודולוגיה לבדיקות אתרים ושימושיות ב-Web: בעיות אופייניות, מה צריך לבדוק ואיך נבצע בדיקת אפליקציות Web וממשקים, נלמד על Web Server - קישוריות לשרת, מדידת זמן תגובה, ביצועים בדיקות עומסים ותכנון בדיקות על אתר אינטרנט בתהליך פיתוח באמצעות כלי ניהול. נעשה שימוש בכלי בדיקות נוספים, כדוגמת: Web, Wireshark, Developers Tools, Fiddler. ובנוסף, נכיר את טכנולוגיות בניית דפי ה-Web הבאים:

HTML5 + CSS3

ביחידה זו נכיר את שפת התגים HTML והעיצוב CSS תוך הבנת פיתוח דפי אינטרנט סטטיים ושילוב העיצוב באתרי אינטרנט. מודול זה כולל נושאים כגון פיתוח דפי אינטרנט סטטיים תוך שימוש בשפת HTML, עיצוב האתר באמצעות CSS, תחביר CSS, כתיבת דף CSS בודד לעיצוב מספר דפי HTML, שימוש ב-CSS מתקדם, כיצד ניתן לעבוד עם מודלים שונים כגון, SPA (Single Page Application) ועוד.

JavaScript

שפת JavaScript הינה הבסיס לטכניקות תכנות מוכרות והיא השפה הנפוצה ביותר בעולם התכנות לצד לקוח. חלק זה כולל נושאים כגון, הכרות עם שפת JavaScript, משתנים, אופרטורים, משפטי תנאי, לולאות, שימוש בפונקציות, שימוש במערכים ושימוש באובייקטים.

כלי בדיקה כדוגמת: Wireshark, Web Developers Tools, Fiddler.

פרויקט מס' 2 - בסיום מודול זה יבוצע פרויקט אינטגרטיבי בנושא בדיקות תוכנה בתחום WEB במתכונת של עבודה עצמית (25 ש"ל)

מודול 7 - בדיקות עולם ה-Mobile באמצעות Appium (10 ש"א)

בחלק זה של המסלול נכיר עולם המובייל וחשיבותו היום בדגש על מתודולוגיה לבדיקות מובייל וההבדלים בין עולם Web והמובייל. נכיר את מערכות הפעלה iOS ו-Android ונסקור את ההבדלים, היתרונות והחסרונות של מערכות אלו כולל עולמות UX, Localization, Network.

פרויקט מס' 3 - פרויקט אינטגרטיבי בנושא בדיקות תוכנה בתחום Mobile במתכונת של עבודה עצמית

מודול 8 - מתודולוגיית Agile (4 ש"א)

הצורך בגמישות הוליד את מתודולוגיות ה-Agile ובפרט את תהליך ה-Scrum המחלק בנייה של מוצר גדול לקבוצות של פיצ'רים קטנים יותר. עקרונות נוספים על פיהם נוהגים ב-Agile הינם שיתוף פעולה הדוק בין צוותי הפיתוח ללקוחות. חלק זה של המסלול, מקנה למשתתפיו הכרות עם הערכים העסקיים מיישום Agile בארגון, לספק למשתתפים כלים שיסייעו להם לבחור את הגישות המתאימות להם.

מודול 9 - תהליך CI/CD (3 ש"א)

CI/CD (Continuous Integration ו-Continuous Deployment) הם תהליכי פיתוח תוכנה זריזים המאפשרים לנו זמן פיתוח קצר בהרבה, שמירת תאימות לאורך זמן, תיקון באגים בזמן קצר בהרבה ועוד. בחלק זה של המסלול נכיר את תהליכי CI/CD ונציג מערכת CI/CD כדוגמת Jenkins.

מבחן מסכם : בדיקות אוטומציה (2.5 ש"א)

הכנה למבחני הסמכה ISTQB (35 ש"א)

במסגרת המסלול תערך הכנה לבחינת הסמכה ISTQB ברמת Foundation Level. יחידה זו כוללת הכנה מעמיקה תוך תרגול עם מבחני עבר.

פרוייקט גמר עצמי להגשה : (40 ש"א)

בחלק זה של המסלול יבוצע פרוייקט גמר אשר יסכם את כל החומר הנלמד.

**שלב ג' - ארכיטקטורת
תוכנה 120 ש"א**

מודול 1 - מבני נתונים אבסטרקטים (15 ש"א)

- נתונים בסיסיים - מספרי, בוליאני, אלפבתי.
- מבנים מורכבים - מערך. מחרוזת, רשומה, קבוצה.
- גישה לנתונים - גישה ישירה, סדרתית, אינדקס, שיטות חיפוש.
- הוספת והוצאת נתון ממבנה הנתונים - מבנה סטטי ומבנה דינמי.
- מיון הנתונים.
- התאמת מבנה הנתונים לסוג הבעיה - שיקולים בבחירת מבנה הנתונים.
- מערכים - הוספת איבר, ביטול איבר, יצוג מערכים חד מימדיים ודו מימדיים בזיכרון, מיזוג מערכים.
- מחסנית - פעולות PUSH ו-POP, מחסנית דינמית, מחסנית מטיפוסים שונים, יישום המחסנית בחישוב ביטויים אלגבריים.
- תור - הגדרה לוגית, פעולות הכנסה והוצאה מהתור, יישום בזכרון סטטי ובזכרון דינמי.
- רשימות מקושרות - הגדרת רשימה מקושרת, הכנסה והוצאה של איברים, יישום מחסנית בעזרת רשימה מקושרת, יישום תור בעזרת רשימה מקושרת.

מודול 2 - אלגוריתמיקה (15 ש"א)

- חיפוש - אלגוריתם לחיפוש במבנה נתונים.
- מיון - אלגוריתם למיון נתונים בסדר מסוים.
- הכנסה - אלגוריתם להכנסת ערך במבנה נתונים.
- עדכון - אלגוריתם לעדכון ערך קיים במבנה נתונים.
- מחיקה - אלגוריתם למחיקת נתון קיים במבנה נתונים

מודול 3 - חשיבה אלגוריתמית (10 ש"א)

בחינה של אלגוריתמים מהיבטים של נכונות, סיבוכיות ומבניות.

מודול 4 - עקרונות מודולריים נכונים (15 ש"א)

● SOLID:

- Single Responsibility
- Open/Closed
- Liskov Substitution
- Interface Segregation
- Dependency Inversion
- DRY (Don't repeat yourself)

מודול 5- עקרונות לכתיבת קוד - Scalability (10 ש"א)

- תכנות ראשוני (Design) - כתיבת מסמך דרישות, ניתוח מסמך דרישות, אנליזה ותכנון.
- יישום מעלה-מטה (Top-Down Implementation) - הפשטה (Abstraction), מודולציה (Modular Design), כתיבת פונקציות קצרות וייעודיות, יצירת ממשק תכנות יישום (Application Programming Interface), כתיבת הערות, בחירת שמות (Naming Conventions), ניידות הקוד, ניטרול שגיאות.

מודול 6 - Design Patterns (7 ש"א)

- Singleton
- Facade
- Factory
- Proxy
- Delegate

מודול 7 - גישה לנתונים - Data Access (4 ש"א)

- private
- protected
- default (package visible)
- public

מודול 8 - עקרונות Access control (4 ש"א)

- Mandatory access control
- Discretionary access control
- Role-based access control
- Attribute-based access control

מודול 9 - Python (30 ש"א)

- התקנת Python, סוגי משתנים, תנאים, לולאות.
- מערכים ומחרוזות.
- פונקציות: הגדרה, קריאה והעברת פרמטרים, רקורסיות.
- מחלקות: הגדרת מחלקות, משתנים, פונקציות, בנאי ויצירת מופעים.
- Modules & Packages, Iterators, Tuple
- שימוש בקבצים: Create files, Write to files, Delete files
- Networking Programing: Socket, Domain, Simple Server, Simple Client, Sending Emails
- Multithreading Programing: New thread, Sync threads, Thread priority

מודול 10 - פרויקט מסכם - ReFactoring (10 ש"א)

בחלק האחרון של המסלול נבצע מחדש את פרוייקט הגמר אבל הפעם בתור QA Architect

שלב א'

מודול 1 - מבוא למחזור חיי בדיקת תוכנה
- היסטוריית בדיקות. - מה היא בדיקה? - מדוע נדרשת בדיקה? - מה הוא Bug? - מחזור חיי ה-Bug - עקרונות בדיקה. - הבטחה ובקרת איכות. - היקף בדיקה. - מתי בדיקה צריכה להתרחש. - בדיקת אילוצים. - תפקידים של בודק התוכנה. - SDLC - מודלים שונים של מחזור חיים. - STLC

מודול 2 - תכנון בדיקה
- אסטרטגיית בדיקות. - תכנון בדיקות. - התאמה אישית של תהליך הבדיקה. - ניהול זמנים. - ניהול תצורה. - ניהול סיכונים.

מודול 3 - עיצוב בדיקה
- תרחישים לבדיקה. - מקרי בדיקה. - נתוני בדיקה. - הפער בין תרחיש הבדיקה לתוצאה. - יצירת מקרה בדיקה לאפליקציה פשוטה.

מודול 4 - טכניקות בדיקה

טכניקות סטטיות:

- חשיבות הבדיקה ב- STLC (Software Testing Life Cycle)
- פעולות בדיקה
- תפקידים ואחריות במהלך הבדיקה
- טכניקות דינמיות:
- טכניקות מבוססות מפרט או קופסא שחורה
- ניתוח ערך גבול
- בדיקת טבלת החלטות
- חלוקה אקוויולנטית
- טכניקות מבוססות נסיון:
- ניחוש שגיאה
- בדיקות גישוש

מודול 5 - רמות וסוגי בדיקות

רמות בדיקה:

- Unit testing
- Integration testing
- System check
- User acceptance tests

סוגי בדיקות:

- Regression testing
- Smoke testing
- Database testing
- Load testing
- Performance testing
- Compatibility testing
- Volume testing
- Capacity testing
- Spike testing
- Crowd testing
- Security testing
- Stress testing
- Usability testing
- Localization testing

מודול 6 - ביצוע בדיקות
• הכרות עם Build & Release • מסמכי שחרור • 'צ'ק ליסט' מקדים • קריטריוני כניסה ויציאה • ביצוע בדיקה

מודול 7 - ניהול Bugs
• מניעת Bug's • גילוי Bug's • Bug's במחזור החיים • הכרות עם RCA (Root Cause Analysis) • זיהוי ורישום חומרים • עבודה "hands on" - למצוא באגים במוצר

מודול 8 - דיווח ושיתוף פעולה בצוות
• STP (Software Test Plan) • STD (Software Test Description) • STR (Software Test Report) • דוחות סגירת בדיקה • בודק ומפתח • שיתוף פעולה קבוצתי • אינטרקציה עם לקוח • מצבי Onshore/Offshore • הקלת האתגרים הנוכחיים

מודול 9 - מדדי איכות - KPI
• סקירת מדדים ומדידות • היתרונות של מדידת בדיקה • מחזור חיים של מדדים • סוגי מדדים

שלב ב'

מודול 3 - מסדי נתונים - SQL
עיצוב ותכנון בסיס נתונים: - ERD – Entity Relationship Diagram - שפת SQL - DML -Data Manipulation Language - DDL – Data Definition Language

מודל 4 - ניהול תצורה והפצת קוד GIT
Introduction - What is Git - About Version Control System and Types - Difference between CVCS and DVCS - A short history of GIT - GIT Basics - GIT Command Line Installing Git - Installing on Windows - Initial setup Git Essentials - Creating repository - Cloning, check-in and committing - Fetch pull and remote - Branching

מודול 5 - Selenium & Frameworks

- Test automation framework
- מרכיבים של Selenium
- WebDriver locators (XPath and CSS, Id,Class ,Data-Hook)
- WebDriver API
- ארכיטקטורת פרויקטי אוטומציה
- Java, Json, XML & Web Services
- Testing REST
- Introduction to Cypress.IO & Puppeteer

מודול 6 - בדיקות אוטומציה בעולם ה-WEB

HTML5 + CSS3:

HTML:

- Editors, Elements, Attributes, Headings, Paragraphs, Styles, Formatting, Comments, Colors, Links, Images, Tables, Lists, Blocks, Id, Iframes, File Paths, Head, Layout, Responsive, Entities, Symbols, Charset, HTML Forms, Media, References.

CSS:

- Syntax, Selector, Colors, Backgrounds, Borders, Margins, Padding, Height/Width, Box Model, Outline, Text, Fonts, Icons, Links, Lists, Tables, Display, Position, Overflow, Float, Inline-block, Align, Navigation Bar, Dropdowns, Attr Selectors, Forms, Website Layout.
- CSS Advanced: Responsive, Grid, References.

JavaScript

- Comments, Variables, Operators, Arithmetic, Assignment, Data Types, Functions, Events, Strings, String Methods, Numbers, Number Methods, Arrays, Array Methods, Array Sort, Array Iteration, Date Formats, Date Methods, Math Methods, Comparisons, Conditions, Switch, Loop's, Type Conversion, RegExp, Exception Handling, Scope, Strict Mode, Arrow Function, Classes, Debugging, Performance, Reserved Words, Versions, JSON, Forms, API, Function's.
- Object's: Definitions, Methods, Constructor, Properties, Prototype.
- DOM: Methods, Document, Elements, HTML, CSS, Animations, Events, Event Listener, Navigation, Nodes, Collections, Node Lists, Browser.
- AJAX: XMLHttpRequest, Request, Response, XML File, ASP, Database, Applications, Examples.
- JSON: Syntax, JSON vs XML, Data Types, Parse, Stringify, Objects, Arrays, HTML, JSONP.
- JS References: JavaScript Objects, HTML DOM Objects.

מודל 8 - מתודולוגיית Agile

- למה Agile?
 - הבעיות והאתגרים
 - כיצד מתודולוגיית Agile מסייעת להתמודדות עם בעיות אלו
 - מה הייתרונות שארגונים שאימצו אג'יל מדווחים עליהם
 - למה אג'יל בארגון שלנו?
- מה זה Agile?
 - עקרונות ה Agile ו ה Lean
 - ההיסטוריה של ה Agile
 - The Agile Manifesto
- Scrum, Kanban and XP - הכרות כללית עם כל אחת מהגישות
 - Scrum - עקרונות ושיטות
 - Kanban - עקרונות ושיטות
 - XP - עקרונות ושיטות
- ניהול הדרישות, המוצר וה Backlog
 - עקרונות בניהול הדרישות
- ניהול האיכות והבדיקות
- מקומות של המנהל בארגון/ פרויקט אג'יל
- מה בין DevOps לאג'יל
- Scrum (או Kanban) בהרחבה
- העקרונות, והתהליכים
- תפקידים, ישיבות וכלים
- ניהול התכולות ותהליך ה Planning
- יצירה וניהול של ה Backlog בהרחבה
- ניהול סיכונים
- ניהול התקשורת והאנשים
- בקרה - כלים שיטות
- ניהול האיכות ותהליכי הבדיקות
- שיפור מתמיד ו Retrospective
- תהליך הטמעת אג'יל בארגון / צוות פיתוח
 - השלבים בתהליך ההטמעה - הטמעה כתהליך שינוי ארגוני
 - ניהול סיכונים בפרויקט ההטמעה



המרכז הבינלאומי
ללימודי הייטק וחדשנות

* 6377

מתקדמים
לקריירה בהייטק



Microsoft Partner
Gold Learning



GLOBAL
UNIVERSITY
SYSTEMS



HIGHQ
מכנים אותך לעולם האמצעי



The University of
Law

London
School of Business
& Finance



HTK



TORONTO
SCHOOL OF MANAGEMENT

קמפוסים בפריסה ארצית:

באר שבע

רחוב האנרגיה 77
פארק ההייטק

ירושלים

רחוב יפו 34

רחובות

רחוב אופנהיימר 5
פארק המדע

תל אביב

ראול ולנברג 36
קריית עתידים