



אבטחת איכות תוכנה מעשית

סילבוס הקורס

גרסה 1.0

נקסוס פתרונות מחשוב

טל': 1-800-10-20-74 פקס: 0722-740-740 www.nexus-it.co.il

1 כללי

1.1 תאור הקורס

בדיקות תוכנה הופכות עם הזמן לחלק הכרחי במאבק לאיכות המוצר. בד בבד עם המורכבות הגוברת של מוצרי התוכנה גוברת הדרישה מצד הלקוחות לקיצור זמני פיתוח; גם התחרות בין חברות פיתוח שונות אינה משאירה מקום לטעויות. הגורמים המתוארים מחייבים תהליכי איכות מקיפים ויעילים, תוך מתן דגש על תכנון אסטרטגי ואפקטיבי לא פחות מאשר על ביצוע הבדיקות עצמן.

המטרה העיקרית של הקורס "אבטחת איכות תוכנה מעשית" היא רכישת ידע בתהליכי איכות ומתודולוגיות מקובלות בתחום הפיתוח, תוך הקניית כלים מעשיים לעבודה כמהנדסי בדיקות בארגוני תוכנה. אנו בנקסוס פתרונות מחשוב מאמינים בהיותם חלק בלתי נפרד המהווה בסיס להכשרה של מהנדסי בדיקות מקצועיים. לכן, חלק ניכר מהקורס יוקדש להיכרות מעשית עם סביבת עבודה של אנשי בדיקות וטכנולוגיות הנמצאות בשימוש רחב בתחום התוכנה.

2.1 מטרת הקורס

מטרת הקורס היא הכשרה של אנשי בדיקות תוכנה, וכוללת בין היתר התמחות בנושאים הבאים:

- היכרות עם בדיקות ואבטחת איכות תוכנה כחלק אינטגרלי בתהליכי פיתוח
- מתודולוגיות, מושגים ושיטות מקובלות בתחום
- יכולת ניתוח מסמכי אפיון מוצר, עיצוב ותכנון תסריטי בדיקות מתוך מסמכי האפיון וכן תכנון הרצתן, תוך שמירה על קשר עקבי ומתועד בין כל השלבים.
- הכרת כלים ייעודיים לניהול תהליכים ומעקב תקלות
- התנסות מעשית בכלי בדיקות ייעודיים וכן כלים נפוצים המלווים את תהליכי הבדיקות
- רכישת ניסיון בבדיקת אפליקציות מסוג שרת-לקוח (client-server)

3.1 דרישות קדם

מתכונת הקורס מכוונת לבעלי ניסיון ולא ידע קודם בעבודה בתחום מחשבים, שכן מדובר על קורס מתקדם. קבלה לקורס מותנית במספר דרישות קדם

- מעבר של מבחן כניסה בציון של 75 לפחות
- הכרה של סביבת עבודה Microsoft Windows ברמת משתמש (הכרת Linux – יתרון)
- ידע באנגלית ברמה של קריאה טכנית לפחות
- תואר אקדמי או השכלה דומה – יתרון
- ידע מוקדם בפיתוח תוכנה – יתרון

4.1 משך הקורס

הקורס מתוכנן ל-180 שעות לימוד אקדמיות (45 דק'), המועברות במתכונת של 3 מפגשים בשבוע, שאורכם 4 שעות כ"א, למשך תקופה של כ-4 חודשים. משך הקורס בפועל יכול להיות מוארך עקב חגים יומי שבתון.

5.1 חומר לימוד

- חומר הלימוד יינתן בשפה אנגלית
- חלוקת חומר תתבצע בצורה אלקטרונית, לפי סדר הנושאים הנלמדים, בסמוך ללימוד הנושא.
- חומר הלימוד יהווה תמצית של החומר הנלמד בשיעור ויוכל לשמש לחזרה על החומר

■ חומר הלימוד לא מהווה תחליף לנוכחות בשיעורים

6.1 משימות לעבודה עצמאית

בנוסף למפגשי ההדרכה, על התלמידים יהיה להשלים פרויקט גמר, המורכב ממספר חלקים.

- אופן ההגשה הינו לפי חלקים, במקביל לחומר הנלמד.
- לכל חלק בפרויקט מועד הגשה סופי, לאחריו העבודות לא יתקבלו.
- כל חלק בפרויקט מהווה יחידה נפרדת, אשר לה ציון נפרד
- לאחר סיום חלק כלשהו, יוצג פתרון מוצע לעבודה, אשר יהווה בסיס אחיד ויחיד לחלקי הפרויקט הבאים. אין להסתמך על פתרון עצמאי של שלב כלשהו בפתרון של שלבי פרויקט מתקדמים.
- ההגשה של הפרויקט חייבת להתבצע בקובץ בפורמט Microsoft Word או Adobe PDF
- אי סדר בהגשת עבודה יכול לגרום להורדת הציון.
- הציון הסופי יחושב כממוצע של הציונים עבור כל חלק נפרד מן הפרויקט
- ציון עבור שלב בפרויקט אשר לא הוגש במועד ההגשה יזן כ- 0.

7.1 תנאי סיום

סיום מוצלח של הקורס מותנה במספר תנאים הכרחיים:

- נוכחות של 80% לפחות¹
- מעבר של מבחן גמר בציון של 85 לפחות²
- הגשה של פרויקט גמר בציון של 75 לפחות³

8.1 תעודת סיום

בוגר הקורס יקבל תעודת סיום קורס רשמית, שתכלול את מתכונת ומשך הקורס ואת ציוניו הסופיים בפרויקט ומבחן גמר.

¹ סטודנט שלא נכח בחלק מן הקורס מסיבה מוצדקת יהיה רשאי להשלימו במועד אחר בתאום מראש עם ההנהלה.

² סטודנט שלא עבר את מבחן הגמר יהיה רשאי לגשת למבחן נוסף בתוספת תשלום

³ סטודנט שקיבל ציון נמוך יותר בפרויקט הגמר יהיה רשאי להגיש פרויקט גמר נוסף ממועד קורס אחר במלואו

2 נושאי הקורס

להלן מוצגת מתכונת מתוכננת לחומר הקורס.

הערה: הנושאים מוצגים לפי סדר כרונולוגי בפיתוח יישומים. מטעמי נוחות הדרכה ותלויות בין נושאים, סדר הלימוד של הנושאים בפועל נתון לשינוי.

1.2 מבוא לבדיקות ואבטחת איכות תוכנת לימוד תיאורטי) 4 שעות

המטרה העיקרית של חלק זה היא היכרות המועמד עם מהלך חיי התוכנה, מקומם של מהנדסי האיכות בתהליך הפיתוח כחלק בלתי נפרד ממנו, וכן מושגי יסוד בסיסיים מעולם התוכנה:

- מושגי יסוד
- מודלים של פיתוח תוכנה ומהלך חיים (Waterfall, Sashimi, Spiral, V model, Agile)
- אבטחת איכות תוכנה כחלק במהלך חיי התוכנה
- גישות מקובלות בבדיקות תוכנה (Black-Box, White-Box)
- פאזות של בדיקות במחזור חיי התוכנה (Unit, Component, Integration, System, Acceptance)
- סוגי בדיקות ומטרותיהם (Functional, Interface, Compatibility, Usability, Performance, Load) (subtypes)
- אסטרטגיה של בדיקות תוכנה: שלבים עיקריים בבדיקות ומעבר ביניהם (SRS, STP, STD, STR,) (Maintenance)
- אספקטים פסיכולוגיים של עבודת QA כקבוצה עוינת

2.2 תכנון בדיקות(לימוד תיאורטי) 8 שעות

חלק זה מטרתו העיקרית הינה הקנייה של ידע בסיסי בתכנון בדיקות בשלבי קדם של פיתוח תוכנה. יש לציין כי נושא זה מובא במתכונת מצומצמת. נדון בהרחבה בתכנון הבדיקות בקורס ההמשך, המיועד לראשי צוותים ואנשי בדיקות מנוסים.

- הכרת המושג והצורך בתכנון בדיקות, הצגת מסמך STP
- הנחיות לתכנון בדיקות מוצלח – מה, מתי ואיך לבצע
- חלוקה לנושאים פונקציונאליים כדרך לתכנון נכון
- הנושאים העיקריים עליהם יש צורך להתמקד בזמן התכנון
- אינטראקציה עם מחלקות אחרות בארגון
- "איך לא לעשות" – דוגמאות לתקלות בתכנון הבדיקה מהעולם האמיתי.

3.2 עיצוב תסריטי בדיקות 20 שעות

על נושא עיצוב תסריטי הבדיקות נדון בהרחבה במשך מספר מפגשים, שכן זהו אחד השלבים הקריטיים למוצר. הנושא מתחלק למספר חלקים, אשר ילוו בתרגול מעשי בשיעורים וכחלק מפרויקט הגמר

1.3.2 פענוח מסמכי אפיון(לימוד תיאורטי) 4 שעות

בחלק זה נדון בהרחבה על מהות מסמכי האפיון כגורם החשוב ביותר במהלך חיי התוכנה, ועל ההשפעה הנחוצה של נציגי איכות בשלבי האפיון.

- הדרך מדרישות הלקוח אל מסמך האפיון הטכני
- הכרת מושגים ושלבי ביניים להתהוות האפיון
- מתכונת מסמך אפיון לדרישות המערכת (SRS)

- סוגי דרישות מערכת (Functional, Non-Functional)
- מבחן השפיות של דרישות מערכת (Correctness, Completeness, Consistency, Testability,)
- (Feasibility, Necessity, Prioritization, Unambiguousness)
- סכנה של דרישות מערכת נרמזות (Implied Requirements)
- תפקידו של מהנדס אבטחת האיכות בשלב הכנת האפיון ואינטראקציה עם גורמים חיצוניים

2.3.2 התנסות מעשית בביקורת מסמכי אפיון (דיון) 4 שעות

בחלק זה נעבוד על הכנה לעבודה בפרויקט גמר. נדון על קשיים המתעוררים בזמן פענוח מסמכי אפיון. המטרה של המועמד היא להגיע למודעות גבוהה בבעייתיות של מסמכי דרישות לא נכונים או לא מלאים בכדי להעריך את חשיבות שלבי אבטחת האיכות הראשונים להצלחתו של פרויקט כולו.

חלק מפרויקט גמר – ביקורת של מסמך אפיון קיים

3.3.2 עיצוב התסריטים כנגזרת של מסמך אפיון (דיון) 8 שעות

נתבונן בדרך המקובלת להגדרת מסמך עיצוב התסריטים כנגזרת של מסמכי אפיון באספקטים הבאים:

- הכרת מושגים בסיסיים בנושא עיצוב התסריטים (STD, Use Case, Test Case, Test Scenario)
- קווים מנחים לעיצוב בדיקה מוצלח (Successful Test Design)
- הגדרה של תסריט בדיקות מוצלח (Successful Test Scenario)
- שיטות בדיקה פונקציונאלית בגישת הקופסה השחורה (Boundary values, error guessing,) (decision tables, equivalence partitions)
- התייחסות לדרישות בלתי פונקציונאליות כחלק מאבטחת איכות של המוצר.
- שמירת עקביות בין פאזות שונות של פרויקט ע"י מטריצת עקיבה (Traceability Matrix)
- דוגמאות לבעיות שכיחות בתהליכים של פיתוח מוצר.

חלק מפרויקט גמר – כתיבת מסמך STD על סמך אפיון קיים

4.3.2 מימוש תסריטי בדיקות בפרוצדורות דטרמיניסטי (דיון) 4 שעות

בחלק זה נדון בהרחבה במעבר משלב התיאוריה ותכנון המוצר לשלב המימוש האחרון לפני הרצת הבדיקות, תוך התבוננות במימוש:

- הכרת מושגים בסיסיים בנושא המימוש (Use Case, Test Case, Test Step, ATP)
- קווים מנחים לניסוח תסריטי בדיקות
- המלחמה בין פרוצדורות מפורטות לבין קווי מתאר – יתרונות וחסרונות של כל שיטה
- שיקולים של ניהול וקלות עדכונים – הכנה של עץ בדיקות ללא שכפולים
- ניסוח פרוצדורות לבדיקות לא פונקציונאליות

חלק מפרויקט גמר – כתיבת מסמך ATP על סמך STD קיים

4.2 הרצה של תוכנית בדיקות ומדידת איכות 16 שעות

נושא זה דן בשיטות מקובלות לביצוע בדיקות בפועל (הרצה), תכנון מחזורי בדיקות וסדרי עדיפויות בהרצה, פענוח תוצאות, מדידות ושערוכי איכות על סמך עמידה בקריטריונים. כמו הנושא הקודם, גם כאן קיימת חלוקה לתחום ההרצה ותחום האדמיניסטרציה, או באם תרצו – טקטיקה ואסטרטגיה.

4 שעות

1.4.2 איתור תקלות תיעוד ומעכב(דיון)

בשיעור זה נדון בצורה מקיפה במציאת תקלות, תיעודן ומעכב אחר תיקון כגולת ה כותרת של תפקיד אבטחת האיכות:

- הכרה של מושגים בנושא (Defect, Fault)
- תקלה כאי עמידה בדרישות אפיון מובנות או נרמזות
- מהלך חיי תקלה ושיתוף פעולה של גורמים מעורבים.
- אנליזה ראשונית ואיסוף מידע לצורך שחזור והבנת התקלה
- הצורך במעכב אחר תקלות פתוחות ויידוא תיקון – תחומי אחריות הבודק

4 שעות

2.4.2 הרצה של תוכנית בדיקות בסביבת מעבדה(דיון)

בשיעור זה נדון באספקטים שונים של הרצת בדיקות במעבדת RND ואיסוף תוצאות:

- מושגים בסיסיים (Branch, Merge, Code Freeze, System Under Test)
- הכנה של סביבת בדיקות ותאורה כסביבה סגורה ובעלת גבולות מדויקים
- מעכב אחר גרסאות תוכנה וגישה לסביבת הבדיקות ע"י צוות לא מורשה
- איסוף מידע מקיף על מערכת תחת בדיקה לצורך פענוח וקורלציה מאוחרת

4 שעות

3.4.2 תכנון של מחזורי בדיקות(לימוד תיאורטי ודיון)

בחלק זה נדון על תכנון וביצוע של בדיקות בעולם האמיתי ובאספקטים נוספים האמורים להילקח בחשבון לצורך תכנון יעיל:

- חשיבות מחזורי בדיקות כהליך בלתי נפרד מאבטחת איכות של מוצר מתפתח
- השפעה של סדר עדיפויות בשלבי הפיתוח על סדר קדימויות בדיקה
- הקמת הסביבה כשיקול בקדימויות הבדיקה, קיבוץ תסריטי בדיקות לפי דרישות קדם
- תקשורת עם גורמים אחרים במהלך הרצת בדיקות כחלק בלתי נפרד מבדיקות יעילות
- מעברים בין מחזורי בדיקות ואנליזה של שערי מעבר (transition criteria)
- משימות הנלוות לסיום מחזור בדיקות

4 שעות

4.4.2 דיווח על התקדמות הבדיקות ודרך להוצאתמוצר אל המדף(דיון)

נתמקד בפענוח מידע סופי באשר מהות הבדיקות ותוצאותיהן, ככלי לשערוך איכות מוצר:

- הכרה של מושגים בסיסיים (Coverage, Exit Criteria, Go-No-Go / QARR)
- נוסחאות חישוב לכיסוי מוצר וחשיבות של עקביות במסמכים לצורך חישוב כיסוי דרישות
- מחלקת אבטחת איכות כשער סופי בדרך של המוצר אל המדף
- התנגשות שיקולי איכות מוצר בגורמים אחרים ומאבקי כוח, אספקטים פסיכולוגיים של "העדר המילה האחרונה"

8 שעות

5.2 בדיקות של דרישות לא פונקציונאליות

נושא זה דורש הרחבה בפני עצמו, שכן האספקטים הלא פונקציונאליים אחראיים לכ-80% של תקלות תוכנה קריטיות הנמצאות בשטח ע"י לקוחות בשנים האחרונות.

1.5.2 בדיקות שרידות והתאוששות (לימוד תיאורטי) 2 שעות

אחד מן האספקטים החשובים ביותר ביצירת מערכות מחשוב היום הוא נגישות גבוהה (קבועה) למערכת ואפשרות להתאוששות מהירה ו"שקופה" במקרה של תקלה קריטית בזמן ריצה. נדון באספקטים אלו:

- הכרה של מושגים בסיסיים (MTBF, MTTR, MDT, availability levels)
- תוצאות של נפילת מערכת שלא עברה בדיקות התאוששות
- נקודות טורפה נפוצות בתכנון ועיצוב מערכות לא שרידות (hardware, backup, database and transaction modeling)
- דרישות לשרידות והתאוששות כפונקציה של יעוד המערכת

2.5.2 בדיקות ממשקים ותאימות (לימוד תיאורטי) 2 שעות

כל אפליקציה חושפת לפחות ממשק אחד לכיוון ישות חיצונית כלשהי, אדם או מכונה אחרת. חלק זה מתמקד בבדיקות של ממשקים אלו בהתאם לסוג הישות איתה התוכנה אמורה לתקשר

- דגשים בבדיקות של ממשקי אדם ⇔ מכונה – usability, design, functionality
- דגשים בבדיקות של ממשקי מכונה ⇔ מכונה – conformance and compliancy, security, error handling
- דגשים לעיצוב ובדיקת ממשקים אינטרמודולאריים במערכות מבוזרות
- חשיפה של ממשקים לאויב לא מתוכנן כנקודת טורפה של המערכת
- מודלים של אפליקציה מסוג לקוח-שרת (Thin/Thick client, clientless) והבדלים בבדיקה של מערכות אלה

3.5.2 בדיקות עומסים (לימוד תיאורטי) 3 שעות

נדון בהרחבה בסוגי בדיקות עומסים נפוצים וכן במטרות העיקריות לביצוען:

- סוגי בדיקות עומסים (Performance, Stress, Longevity)
- יתרונות וחסרונות של בדיקת עומסים פונקציונאלית
- כללי אצבע להגדרת תסריטי בדיקות עומסים
- דגימת מידע אודות מצב המערכת במהלך בדיקות עומסים ככלי לפענוח תוצאות הרצה
- הכרת מושגים ונוסחאות חשובות לבדיקות עומסים (TPS, concurrency, success ratio)
- קריטריונים לקביעת שיעור הצלחה של בדיקות עומסים

במהלך הקורס נלמד מס' כלים אוטומטיים, ונטפל בנושא של בדיקות עומס גם באספקט המעשי.

4.5.2 אבטחת איכות של מאפיינים בלתי מובהקים של מערכות תוכן(היון) 1 שעות

למרות השקעה רבה ככל האפשר במסמכי אפיון ראשוניים של מערכת מחשוב, תמיד יהיו חסרים מאפיינים חשובים יותר או חשובים פחות. בשיעור זה נדון על מקומות חשודים בהם נפוצות תקלות חמורות ועל דרכים לאתרן:

- חשיבות של רישומי מערכת תקינים (log files) ומאפייני מערכת רישום טובה
- בדיקות "חקר" חופשיות ללא תסריטי בדיקות מוגדרים מראש ככלי למציאת תקלות מורכבות ולפענוח תקלות ידועות.
- דגש על הנדסת אנוש ונוחות שימוש – עקרונות "הלקוח הראשון, הלקוח האחרון"

חלק מפרויקט גמר – אפיון של ממשקים לפי סוג על סמך מסמך אפיון קיים
אפיון של בדיקות עומסים לאפליקציה נתונה .

6.2 ניהול סיכונים – QA כחלק מחיי הפרויקט (דיון) 8 שעות

פיתוח תוכנה נחשב אחד התחומים הבעייתיים ביותר מבחינת כמות תלויית בגורמים חיצוניים, לדוגמה – הלקוחות, ספקי מערכת הפעלה, טכנולוגיות freeware וכד'. בחלק זה נדון בהרחבה על הסיכונים הצפויים ועל ניהולם.

- זיהוי והערכת סיכונים, תכנון להימנעות מסיכונים
- שיטות לחישוב והערכת סיכונים
- תכנון וניהול סיכונים – בקרה ומעקב, הרמת דגלים ותיעוד כחלק מהתמודדות עם סיכונים
- דוגמאות לנזקים של ניהול סיכונים כושל
- חוק של ברוקס (Brooks Law)
- התמודדות של מחלקת QA כאחרונה בתהליך פיתוח תוכנה עם סיכונים מצטברים

7.2 גישות ושיטות מתקדמות באבטחת איכות תוכנה (דיון) 4 שעות

החלק הנוכחי דן באספקטים מתקדמים של אבטחת איכות תוכנה כחלק מארגונים בעלי אופי מסוים, וזאת על מנת לאפשר למהנדס איכות להתאים את עצמו בצורה האופטימאלית לעבודה בארגון מסוים.

- מעמד QA בשרשרת ניהול של ארגון, משקלו בקבלת החלטות
- אספקטים פסיכולוגיים של מעמד ה-QA בארגונים שונים, השפעת בגרות הארגון על המעמד
- אפקטיביות של QA – מחיר מציאת תקלה בשלבים שונים של פיתוח
- קביעת שערי מעבר בין מצבים שונים בתהליך הפיתוח ואכיפתם כחלק בלתי נפרד מאבטחת איכות
- התאמה למודל הפיתוח בארגון – יתרונות וחסרונות של שיטות Agile, XP והדרך להתמודדות עמן.

8.2 בדיקות תוכנה אוטומטיות 20 שעות

בחלק זה נדון על היתרונות והחסרונות העיקריים של אוטומציה בארגוני תוכנה, וכן נלמד מספר שיטות מקובלות לביצוע בדיקות אוטומציה בדגש על שיטות אוטומציה עיקריות, וכן נכיר כלים אוטומטיים נפוצים

1.8.2 עקרונות אוטומציה (לימוד תיאורטי) 4 שעות

בשיעור זה נדון בתיאוריה של הקמת מערך בדיקות אוטומטי

- יתרונות וחסרונות של בדיקות אוטומטיות
- כדאיות הקמה של מערך אוטומציה בשלבים שונים של פרויקט
- אוטומציה של בדיקות פונקציונאליות
- שיטות נפוצות לביצוע אוטומציה (Record/Playback, Stateful Automation)
- אוטומציה של בדיקות עומסים – יתרונות וחסרונות
- גרסיה ופרוגרסיה – האם רובוט יחליף בן אדם? (דיון)
- סביבות פיתוח מונחות-בדיקות (Test-Driven development)

2.8.2 הכרה של כלי בדיקות אוטומטיים מתרגול מעשי 16 שעות

בסדנה זו נכיר מספר כלי אוטומציה בהתאם ליעוד הקורס

חלק מפרויקט גמר – הכנת תסריטי בדיקות אוטומטיים לתוכנה קיימת, ע"ס התנהגות ומסמכי עיצוב

9.2 כלים ייעודיים לניהול ומעקב תהליכים 16 שעות

1.9.2 HP Quality Center (תרגול מעשי) 12 שעות

בסדנה זו נלמד את עקרונות הפעלת מערכת, בדגש על יישום מתודולוגיות QA בסביבה ממוחשבת ואכיפת תהליך על ידי מסגרת חיצונית. כמו כן נתרגל פעולות עיקריות

- מבנה היררכי של מודולי התוכנה (Requirements, Test Plan, Test Lab, Bug Management)
- מושגים בסיסיים בעולם של Quality Center (Test Tree, Test Case, Test Step, Run, Coverage)
- עקיבה בין מודולי המערכת (traceability) ככלי חכם לניהול סיכונים
- מאפיינים של ישויות
- שימוש בכלי דיווח של Quality Center למדידת איכות מוצר
- ניהול ואדמיניסטרציה של המערכת, יצירת ישויות חדשות ושינוי תהליכים

חלק מפרויקט גמר – הכנה של תוכנית בדיקות מפורטת ע"פ מסמך עיצוב קיימים

2.9.2 כלי ניהול תקלות – Bug Trackers (תרגול מעשי) 4 שעות

בשיעור זה נדון על כלי ניהול תקלות, בדגש על מערכת ניהול IBM Rational ClearQuest

- מאפיינים של מערכת ניהול טובה
- ניהול מצבים (State Management) של תקלות ומעברים ביניהם
- מטרות והרשאות של גורמים המעורבים במהלך חיי תקלה
- יישום של העקרונות הנלמדים במערכת IBM Rational ClearQuest

01.2 כלים נלווים למהלך בדיקות וטכנולוגיות נפוצות לימוד ותרגול 76 שעות

בחלק זה של הקורס נתמקד בטכנולוגיות נפוצות המופיעות ברוב המוחלט של מערכות אנטרפרייז, אשר ללא הכרתן לא יוכל מהנדס אבטחת איכות לבצע את תפקידו בצורה מהימנה. העברת המידע תבצע בצורה של מתן רקע תיאורטי ויישומו באמצעות כלים ייעודיים נפוצים.

1.01.2 היכרות עם סביבת עבודה במערכת הפעלה Linux (תיאורטי, מעשי) 30 שעות

בסדנה זו נכיר את היסודות של מערכות הפעלה מסוג Unix ואת הכלים הבסיסיים לעבודה בממשק טקסטואלי. במסגרת הסדנה יועבר גם חומר כללי על תורת מערכות הפעלה ברמה הנדרשת להבנה של תהליכים. הסדנה תועבר על מערכת הפעלה Fedora Core 8.0

- מושגים בסיסיים במערכות הפעלה (Process, Thread, Address Space, Shared Memory,) (Interprocess Communication)
- עבודה במצבי מערכת (Kernel Mode, User Mode)
- מצבים של התנגשות בין תהליכים (Starving, Mutual Exclusion, Race Conditions)
- עבודה במערכת הפעלה מרובת משימות (Multitasking)
- מבנה היררכי של מערכת ההפעלה UNIX
- ניהול משתמשים וקבוצות (/etc/passwd, /etc/group), ניהול הרשאות של קבצים
- פקודות בסיסיות להתמצאות בסביבת עבודה, עבודה עם מעבד תמלילים VI
- הגדרות רשת ואדמיניסטרציה של שרתים (Services: HTTPD, VSFTP, XINETD)

- מושגי עומס רגעי וממוצע, Load Average, וניצול משאבי מערכת
- מבוא לתכנות סקריפטים – משתני סביבה, לולאות, בדיקת תנאים לוגיים, error level

2.01.2 רשתות תקשורת TCP/IP 30 שעות

המטרה של נושא זה היא להביא את המועמד להיכרות עם מושגים כלליים ברשתות מחשבים, תוך התנסות במעבדה ויישום מעשי של הידע הנרכש באמצעות שימוש בכלים ייעודיים

- מודל 7 השכבות של OSI ויישומו בפרוטוקולי TCP/IP
- פרוטוקול IP (Addressing Scheme, Classless Routing, Subnetting & Supernetting)
- פרוטוקולי UDP, TCP, וההבדלים המהותיים ביניהם
- היכרות עם פרוטוקולי אפליקציה נפוצים בדגש על HTTP כבסיס לטכנולוגיות חדשות
- היכרות של ציוד בדיקה לתקשורת נתונים (Sniffers, Traffic Analyzers)
- אמצעי תקשורת IP מתקדמים ומטרוטיהם (Proxy, Firewall, Load Balancer, VPN)
- התנסות מעשית בציוד רשת

3.01.2 בסיסי נתונים -ליבה של מערכת תוכנה 8 שעות

בסדנה זו נתמקד במבנה בסיסי נתונים ושפת SQL

- תיאוריה של בסיסי נתונים – המודל הטבלאי
- אלגברה של יחסים ונורמליזציה של בסיסי נתונים
- מפתחות, אינדקסים
- שפת SQL סטנדרטית – סוגי נתונים, מאפייניהם, DDL, מבנה של שאלות בסיסית
- התנסות במערכת MySQL Server, MySQL Administrator
- שימוש בשפת SQL לצורך בדיקות תוכנה

חלק מפרויקט גמר – הכנה של אפיון בסיסי נתונים למערכת לפי דרישות קיימות

4.01.2 מבוא לשפת XML 4 שעות

בחלק זה נדון על מושגי היסוד בהבנת XML

- יסודות Markup Language
- כללי כתיבה של מסמך XML
- וולידציה של מסמך XML קיים
- התנסות בסיסית באפליקציה SoapUI

5.01.2 סביבות ווירטואליזציה של חומרה 4 שעות

בשיעור זה נתמקד על ווירטואליזציה של חומרה לצורך הגברת אפקטיביות עבודה תוך חסכון משמעותי בתקציבים, על בסיס VMware Server. נתרכז בנושאים הבאים

- מושגים בסיסיים בעולם הווירטואליזציה (Host System, Guest System, Virtual Hardware)
- בנייה, שדרוג ועדכון של מכונות ווירטואליות, בניית מפרט חומרה ווירטואלי בהתאם לצרכים
- ניהול של תמונות מצב (Snapshot management) ושימוש בהן בסביבת עבודה של QA
- שכפול וגיבוי של סביבות מרובות לצורך ייעול תהליך ה-Maintenance של גרסאות מוצר קודמות

לפרטים נוספים ולהצעת מחיר יש ליצור קשר עם משרדי החברה

טלפון: 1800-10-20-74

פקס: 0722-740-740

דוא"ל: info@nexus-it.co.il

או לפנות לאתר האינטרנט של הקורס: www.nexus-it.co.il