



שם הקורס בעברית: הנדסת איכות תוכנה

שם הקורס באנגלית: Software Quality Engineering

מספר הקורס: 372-1-3501

מבנה הקורס: הרצאה: 3 ש"ש, תרגיל: 1 ש"ש, מעבדה: 0 ש"ש, סה"כ: 4 ש"ש

נקודות זכות: 3.5 נק"ז

שם המרצה: ד"ר אחיה אליסף achiya@bgu.ac.il

סמסטר א תשפ"ד

תיאור הקורס:

קורס זה סוקר את הרעיונות והעקרונות המרכזיים של הנדסת איכות תוכנה. הקורס יסקור תהליכי איכות תוכנה, מדדי איכות תוכנה, גישות שונות לבדיקות תוכנה ויצג כלים לתמיכה כוללת בתהליכי הפיתוח. דגש רב יינתן לסוגי הבדיקות השונים וליתרונות והחסרונות שלהם. העבודות בקורס יהיו על סוגי הבדיקות ויבדקו את ההבנה התיאורטית והמעשית בנושא זה.

מטרות הקורס:

- הסטודנטים יכירו תהליכי איכות תוכנה ותקני איכות בינלאומיים ומדדי איכות תוכנה מקובלים.
- הסטודנטים יבינו את חשיבות התחום כמרכיב בפיתוח ותחזוקה של תוכנה.
- הסטודנטים יבינו את השיטות שונות לבדיקות תוכנה, ומתי ואיך להשתמש בכל שיטה.
- הסטודנטים יידעו איך לגשת למשימת הבדיקה, איך לפרק את המשימה לתתי משימות, איך להתאים שיטות בדיקה לכל בעיה, ואיך להפעיל כל שיטה.
- הסטודנטים יבינו כיצד ניתן להעריך איכות של מערכת בדיקות ויתנסו בזה בעצמם.



תנאי קדם:

הנדסת תוכנה:

תכנות מערכות: 202-12031

372-13401: ניתוח ועיצוב מערכות להנדסת תוכנה הנדסת מערכות מידע

הנדסת מערכות מידע:

תכנות מתקדם: 37212102

37213801: ניתוח ותיכון מערכות תוכנה (במקביל)

דרישות הקורס והרכב הציון:

1. התלמידים נדרשים ללמוד את נושאי הקורס עפ"י ההרצאות בכתה וחומר הקריאה, ולהגיש את עבודות הבית במועדים שיקבעו לכך.
2. יש לקבל ציון עובר בכל אחד מרכיבי הציון על מנת לעבור את הקורס.
תרגילי הבית – 30%
מבחן פרונטלי באורך של שתיים 70% –
3. במידה ויתקיים ניסוי: יינתנו עד 5 נקודות בonus למשתתפים בניסוי

רשימת קריאה (ביבליוגרפיה):

1. Software Testing and Quality Assurance / Naik, Tripathy
2. Test Driven Development / Kent Beck
3. Metrics and Models in Software Quality Engineering/ Kan



תכנית המפגשים (נתון לשינויים):

| Week | Lecture | Practical Session | Assignment |
|------|--|----------------------------|---|
| 1 | Introduction: SQA research & the software crisis | | |
| 2 | Introduction: Testing types & quality metrics | UT – intro | Publish 1. UT • |
| 3 | Unit testing | UT – isolation | |
| 4 | Control flow & symbolic execution | Control Flow | Submit 1. UT • Publish 2. Control Flow • |
| 5 | Integration Testing | Symbolic Execution | Submit 2. Control Flow • Publish 3. Symbolic Execution • |
| 6 | CTD | Selenium in JUnit | Submit 3. Symbolic Execution • Publish 4. Selenium + Cucumber + CTD • |
| 7 | Acceptance testing & MBT | Cucumber | |
| 8 | Story-based Testing | GraphWalker | Submit 4. Selenium + Cucumber + CTD • Publish 5. Story-based Testing • |
| 9 | Data flow & domain testing | Story-based Testing | |
| 10 | System tests | DataFlow | Submit Story-based Testing • |
| 11 | SQA Processes – CMMI & Review | Rehearsal function point 1 | |
| 12 | AI-Based SQA & Testing AI-based systems | function point 2 | |
| 13 | Rehearsal | Rehearsal | |

* SQA = Software Quality Assurance

* UT = Unit Testing

* CTD = Combinatorics Test Design

* MBT = Model-based Testing