

הנחיות לביצוע פרויקט-ניהול פרויקטי תוכנה

במסגרת החלק היישומי של הקורס, יתבצע פרויקט פיתוח תוכנה בגישת OO, בשפת פייטון או Java, שפות נוספות בתיאום עם המתרגלים. כל תהליך הפיתוח וניהול הפרויקט יתבצע באמצעות GitHub ו JIRA, תוך סנכרון ביניהם. המטרה לפתח את התוכנה בתהליך פיתוח אגילי בשיטת סקראם ובנוסף לישם עקרונות CI/CD המדמה סביבת פיתוח בתעשייה, לאורך 4 ספרינטים (הרביעי לצורכי אינטגרציה וליטוש). להלן דרישות כלליות לגבי הפרויקט. לכל דרישה יהיה ביטוי בשקלול הכולל של הפרויקט.

דרישות כלליות:

1. הגדרת כל הדרישות **כסיפורי משתמש** ו **Epics** ב **PBL** (Project BackLog) בתחילת הקורס- תכולה והיקף יאושרו על ידי צוות הקורס (שבוע 2). לכל ספרינט יוגדר **מראש** SPBL (Spring Back Log), כלומר תכולת הדרישות לאותו ספרינט (שבוע 2). יש להקפיד ולסיים את המתוכנן מבלי להעביר משימות לספרינט הבא. לכל סיפור משתמש יוגדרו משימות **tasks** ויושמו לאנשי הקבוצה על ידי הסקראם מאסטר. הסטאטוסים הקיימים Done, Design review, In progress, Product Backlog. לכל דרישה יש לוודא כיסוי של עיצוב ובדיקות- ראה סעיף 6.
2. לכל ספרינט יהיה **burn down chart** כך שלאורך ההתקדמות יוספו הספרינטים לאותו קובץ לניהול ההתקדמות
3. הגדרת **בעלי תפקידים** – Jira.
4. לכל ספרינט יהיה scrum master שונה, על פי החלטת הקבוצה. כל אחד מחברי הקבוצה יאייש תפקיד נדרש כגון DBA, מפתח, עיצוב, לקוח בכל ספרינט. בכל ספרינט משחק התפקידים ישתנה כך שכל חבר בקבוצה יתנסה בתפקידים השונים.
5. כל צוות מחויב **להיפגש לפחות פעמיים** בשבוע ולדון על ההתקדמות, הבעיות, הפתרונות, מה עושים כעת ומה מתוכנן להמשך (טווח קצר). סיכומי פגישה יוגשו בפגישה עם המתרגל. בהתאם לעדכן את התכנון מול הביצוע. תיעוד המפגשים יוגש בתום הספרינט.
6. בכל ספרינט יתבצע **תכנון, review, ניהול סיכונים עיצוב, פיתוח ובדיקות**- יתועד ב Jira באמצעות הגדרת משימות מתאימות.
7. מחזור חיי הפרויקט יכלול הגדרת מטלות **tasks** לכל סיפור משתמש (לפחות אחת). לכל מטלה יהיה אחראי שיוגדר על ידי ה scrum master. שלבי התקדמות של מטלות. הסטאטוסים הקיימים :
To do, In progress, Done
8. לכל מטלה הצוות אחראי להגדיר ב GIT בדיקות מסודרות ברמת יחידה ותת מערכת. בסוף תתבצע גם בדיקת מערכת ואינטגרציה. השבוע האחרון יוקדש למקצה שיפורים ובדיקות סופיות.
9. יש לקשר את הקוד המתאים לכל סיפור משתמש.

10. בסוף כל ספרינט יש להגיש עם מסמך סיכום (ייצוא Jira), burn down chart, מסמך בדיקות ברמת יחידה ותתי מערכת המתאר את הבדיקות הנדרשות לפונקציות השונות, ואת תוצאות הבדיקה כפי שהתקבלו בהרצה. סטאטוס ההרצה יופיע במטלה ב GIT. פורמט הבדיקות יכול לכלול את החלקים if, when, then.
11. תוכנית עבודה תוצג ב Roadmap שבJira.
12. בפגישת התכנון (בתחילת כל ספרינט) עם המתרגל, ה scrum master אחראי להגדרות מסודרות של כל המטלות בספרינט כולל השמה לסטודנטים והערכת זמנים. בפגישה יעברו על התכנון והמתרגל יאשר או יעדכן במידת הצורך. יש להקפיד להגיע מוכנים לפגישה.
13. בפגישת הסגירה (בתום כל ספרינט) עם המתרגל (sprint summary meeting retrospective), ה scrum master אחראי לסגירה מסודרת של כל המטלות בספרינט כולל עדכון של תוצאות הבדיקות- מראש. סיכום בעיות, מצב נוכחי ותוכנית להמשך. יש להגיש סיכום לכל ספרינט (2-3 עמודים) לתיבת ההגשה. הסיכום יכול ל-3 תתי סיכומים שהוצגו בפגישות השבועיות במהלך הספרינט (ראה סעיף 4).
14. Check in של הקוד בסביבת הפיתוח יקושר לסיפור משתמש\issues, כך שלכל דרישה יהיה כיסוי של קוד. כל חברי הצוות ייקחו חלק שווה בפיתוח.
15. בכל ספרינט יתבצע עיצוב של תרשימי UML, על פי ההנחיות שיתפרסו בהמשך ובתרגול (יתבטא בקישור ל Git). יוגש כחלק מהמסמכים בסוף הספרינט.

תיאור הפרויקטים לפי הנושאים הבאים. ההנחה היא שכל קבוצה נרשמה דרך הקישור בטופס ודירגה את הנושאים מהעדיפות הגבוהה לנמוכה ביותר. שימו לב כי אנו מצפים לחשיבה יצירתית על מנת לפתח את המערכות על בסיס ההגדרות הכלליות המופיעות בהמשך.

לכל פרויקט יש ליישם:

1. לפחות 3 משתמשים.
2. בסיס נתונים: כגון SQL Server, MongoDB.
3. לפחות 36 סיפורי משתמש (לפחות 12 לכל משתמש).
4. לכל משתמש לפחות 6 שאילתות שהוא יכול לבצע
5. לפחות 2 הורשות.
6. על המערכת להיות יעילה ככל האפשר, כאשר המודולים הנבחרים מסונכרנים מבחינת המידע במערכת (לדוגמא: בנושא 1 יש לתכנן את שיבוץ הדרישות בתוכנית עבודה לפי סדר חשיבותן שמוגדר בעת יצירת ספיון משתמש).
7. כל מודול מייצג מערכת, הכוללת ממשק ניהול.
8. ניתן להוסיף רעיונות שלדעתכם מיעילים את המערכת (מודולים, משתמשים וכו').
9. דרישות המערכת יחולקו ל-4 ספרינטים, כאשר 3 ספרינטים באורך של כחודש והספרינט הרביעי הוא באורך של שבוע לצורך ביצוע תיקונים. לפי ההנחיות שלעיל.

10. יש לפתח את המערכת כך שתתמוך בלפחות שני תהליכים מקצה לקצה (E2E). הכוונה להגדרת תהליך הבנוי מרצף של פעולות המאפשרות לבצע משימה מורכבת.

11. פרויקטים המשלבים שפות פיתוח חדשות (כגון מטלב), חומרה ופתרונות שמחוץ לקופסא-יתבטא בתהליך הערכתם.

אפשרות 1- ניהול פרויקט מערכת לניהול ייזום ואפיון תוכנה

נוהל מפת"ח הוא מסגרת מתודולוגית לפיתוח תוכנה כולל היבטים ארגוניים ופרויקטליים של התוכנה. התבניות המוצעות כיום לשלב ייזום ואפיון הפרויקט מכילות מידע רב שלא בהכרח נדרש באופן מלא. כמו כן יש כפילות בין התבניות ולעיתים יש צורך בהתאמה לפרויקטים מסוגים שונים. עליכם לפתח מערכת שתפשט את התהליך של אתחול הפרויקט, כולל הגדרת דרישות כך שניתן יהיה לנהל את התהליך באופן יעיל יותר. תכן לא כלול ניתן למצוא חלופה לנוהל מפת"ח הישראלי על ידי תקנים/נהלים בינלאומיים.

אפשרות 2- תוכנה/אפליקציה לצרכים מיוחדים

פיתוח רעיון לטובת שיפור איכות החיים של אנשים עם מוגבלויות. ניתן לבחור בלקויות שונות (עיוורון, אוטיזם, חרשות, פיגור, CP, ניוון שרירים, וכו'). כמו כן הפתרון יכול להיות מזווית של איבחון, תפקוד שגרתי, טיפול וכו'. על מנת להגדיר את הצורך/בעיה יש למצוא מקורות אקדמיים/מקצועיים רלוונטיים.

אפשרות 3- פיתוח רשת חברתית

המערכת תקשר בין אנשים ותקדם רעיון מסוים תוך התבססות על הקשרים החברתיים שנוצרו ברשת. יש להגדיר את המטרות של הרשת אותן תקדם.

בהצלחה ועבודה פורייה !