

פרויקט משימה 5 – הקמת תשתיות – ZFR

מטרת המשימה

חברת סבב"ה מקפידה על פיתוח איכותי ומבוקר ולכן דורשת ניהול מסודר ומונחה מטרה וכן שימוש בכלי הנדסת תוכנה לניהול הפרויקט ובקרת גרסאות הקוד. בנוסף הוחלט שהפיתוח המתאים להמשך הפרויקט הוא פיתוח בסבבים (ספרינט \ איטרציה). מסמך זה מגדיר את אופן הפיתוח הנדרש להמשך הפרויקט. המשימות הבאות מחולקות לסבב 0 שבו תכינו את התשתיות להמשך העבודה ועוד מספר סבבי פיתוח עד לסיום הקורס. סבב 0 נקרא גם לפעמים Zero Feature Release מכיוון שהוא מתמקד בהתארגנות ניהולית ותשתיתית ללא אספקת מוצר. כל סבב ייחשב גם כאבן דרך ולכן תמורתו תקבלו כבר תשלום (כנקודות בציון הפרויקט). בסבב 0 תכינו: (1) רשימת סיפורי משתמש ממוינת לפי צרכי הלקוח ומחולקת לפי סבבים. (2) השלמת ההכרות עם כלי הפיתוח הנדרשים: א. הצגת אב טיפוס ב. תחילת שימוש במערכת בקרת גרסאות ג. תיעוד ראשוני. (3) תכנון סבב ראשון בשיטת סקראם. בשאר הסבבים תספקו גרסאות הולכות ומתקדמות של המוצר.

לניהול המשימות נשתמש במערכת [github/issues](https://github.com) ובלוח ויזואלי בשיטת קנבן huboard.com. הקוד יאוחסן כמובן ב-github ע"י שימוש ב-git.

תוכן

1.....	מטרת המשימה
2.....	תאריכים והגשות
2.....	1. רשימת מאפיינים (סיפורי משתמש) – Product Backlog
2.....	כללי
2.....	שלבי העבודה (ראו תמונה)
3.....	2. בקרת גרסאות – Version Control
3.....	3. תיעוד המוצר
4.....	4. סקר ZFR
4.....	5. תכנון סבב 1
4.....	6. אב טיפוס

תאריכים והגשות

עד להרצאה הבאה הגשת קישור לעמוד ZFR שמתאר את מצב הפרויקט.

ציונים

הניקוד לפי איכות ההגשה

1. רשימת מאפיינים (סיפורי משתמש) – Product Backlog

כללי



עליכם ליצור טבלת מאפיינים **משויכת, מתועדפת ומוערכת**. המקורות לרשימה זו הם בעיקר סיפורי המשתמש שהכנתם בשלב הדרישות. ישנן מערכות ייעודיות לניהול דרישות ומאפיינים אך אנו נסתפק בהזנת המאפיינים למערכת המשימות של github.




הערכה: כמעט בלתי אפשרי להעריך מאמץ הנדרש למשימה שעוד לא עשינו אף פעם. מצד שני כל מהנדס תוכנה נדרש לתת הערכות. מה עושים? בשלב ראשון מכניסים משימות בסיסיות עם הערכה כללית (באמצעות נקודות-סיפור). לפני כל סבב מנסים לחלק למשימות קטנות יותר עד לרמה שעבודה כבר אפשר לתת הערכה גסה לגבי יחסיות הזמן (בשעות) הנדרש לכל משימה. מתעדים את ההערכות שניתנו ומזינים זמני עבודה כדי שבהמשך נוכל לשפר אותן על סמך הניסיון שנרכש בהערכה.

שלבי העבודה (ראו תמונה)

1. בהסתמך על התכולות שהגדרתם בשלבי הדרישות והתיכון, הצוות מתכנס לתכנן את המשימות ואת קביעת תאריכי הפתיחה והסיום לכל סבב פיתוח. אתם נדרשים לסבב 0 של תכנון והתארגנות באורך של עד שבוע אחד ועוד כארבעה סבבי פיתוח של כשבועיים, כך שתאריך הסיום של הסבב האחרון הוא היום של ההרצאה האחרונה בקורס. הסבבים צריכים להיות באורך קבוע, לפי לוח"ז הקורס.
2. מנהל אתר הפרויקט (המאגר) מכניס במערכת המשימות אבני דרך - אחת לכל סבב מתוכנן (בתפריט הימני באתר: Issues -> Create a new milestone -> Milestones)
הזנת פרטים: שם משמעותי (למשל Iter0-ZFR, Iter1-XXX-MVP), תיאור ותאריך יעד -> (Create).

3. בנוסף נעבוד עם לוח ויזואלי שיעזור לראות את מצב המשימות תוך כדי עבודה. נכנסים ל huboard.com/, מבצעים כניסה (ע"י הכפתור allow public/private access ומאשרים להשתמש בנתוני שם המשתמש שלכם ב-github), בפעם הראשונה תתבקשו לאשר הוספת תוויות לניהול הלוח (הלוח מוסיף תוויות לעדכון מצב העבודה על כל משימה). אתם מתבקשים להוסיף קישור ללוח שלכם מעמוד הפרויקטים באתר הקורס. ניתן גם להוסיף דרך github issues תוויות נוספות לפי הצרכים שלכם (Issues <- labels: שם, צבע <- Create)). הוסיפו לפחות תוויות המסמנות את סוג המשימה, למשל: סיפור משתמש\מאפיין, באג, שיפור וכדו'.
הערה: אפשר להשתמש במערכת חלופית כדוגמת Assana, PivotalTracker, gissues, Mingle, Trello ובלבד שהיא מקושרת לשינויים והמשימות במאגר github שלכם ועונה על הדרישות המפורטות כאן (התייעצו איתנו אם צריך).
4. הצוות מחליט על סיפורים ומשימות בפרויקט (לפי כל מה שהכנתם עד כה, עם דגש על תרחישי שימוש וסיפורי משתמש, אך גם תשתיות והכנות נדרשות הידועות כרגע) ולכל משימה מכניסים אותה למערכת (New Issue <- Issues), עם שם, אחראי (Assigned – במידה וכבר ידוע), שיוך לסבב (Milestone) ותוויות (למשל: דרישה, סיפור משתמש, משימת פיתוח, באג).
בטקסט של כל משימה יש לציין לכל אחת עת העדיפות שלה והערכת הקושי שלה (אפשר להסתפק במערכת פשוטה של 1-3, או לסמן באמצעות תוויות משלכם). טקסט זה גם יכול לשמש אתכם לתיאור תת-משימות ולא תרחיש בדיקה עבור כל סיפור. ראו [כאן](#) איך אפשר להשתמש ב-markdown לסימון פריטים שבוצעו.

2. בקרת גרסאות – Version Control


 במשימה זו עליכם להדגים שהכנסתם כלי בקרת גרסאות לשימוש עבור הפרויקט והוא מכיל כבר שלד של קבצי הפרויקט ולפחות את אב הטיפוס.

הצעדים הנדרשים:

1. יצירת שלד קוד בסביבת העבודה שלכם והוספתו למערכת בקרת הקוד – git (אם יש לכם רק את אב הטיפוס ואתם לא מתכננים להשתמש בו – העלו אותו בתוך תת-ספריה באופן זמני)
2. אופציה: הוספת תת-ספריה עבור מסמכי הפרויקט (SDF – Software Document Folder) וקבצים נוספים שתנוהל במערכת הגרסאות גם כן.
3. דחיפת הקוד למאגר של הפרויקט ב-github (בשלב ראשוני זה מומלץ להשתמש בלקוח שמפשט את התהליך, למשל eclipse egit, github for windows – ראו עמוד הכלים בויקי הקורס).

הערה: הדרישה להמשך הקורס היא שקוד יישמר על יד מי שכתב אותו – ייתכן ניקוד לפי גרף פעילות.

3. תיעוד המוצר

 עליכם לעדכן ולהרחיב בויקי הפרויקט את דפי התיעוד למשתמש (User Manual) ולמפתח (בד"כ readme.md).
להלן הגדרות למידע שמצופה בד"כ מתיעוד. בשלב זה אין צורך בהכרח בכל הפרטים אלא בגרסאות ראשוניות של דפים אלו.

התיעוד למשתמש צריך לכלול:

- פרטים כללים על המוצר

- כיצד להתקין (אם בכלל) ולהריץ את המוצר
- פרטים ראשוניים על אופן השימוש במוצר.
- כיצד לדווח על באג שמתגלה (למשל כתובת מייל, פתיחת נושא\באג במערכת המשימות וכדו')

תיעוד זה יכול להפוך בהמשך למדריך למשתמש, במידה ואתם כבר מכינים מדריך במסמך נפרד, תנו קישור אליו.

התיעוד למפתח נועד עבור מפתחים חדשים שמצטרפים אליכם לפיתוח, לבדוק אותו או להשתמש בו למוצר אחר. הדף הוא הרחבה של מה שכבר הכנתם בשלב אתחול הפרויקט.

הוא צריך לכלול:

- כיצד להשיג את הקוד (גרסה אחרונה וגם כתובת המאגר – במידה ומנוהלים במקום אחר)
- הערה בדבר רישיון הפרויקט (למשל קוד פתוח BSD עם זכויות מסוימות למכללה)
- פרטים כלליים על מבנה הספריות והקבצים בפרויקט
- כיצד לבנות (להדר) את הפרויקט
- כיצד לבדוק אותו (למשל הרצת חליפת בדיקות) – נראה בהמשך.
- כיצד להכין גרסה חדשה (סוג קובץ ארכיב ומיקומו, מספור גרסאות, איכות נדרשת מגרסת שחרור וכדו')
- כיצד להגיע לרשימת הבאגים וכיצד ניתן לקבל משימה לעבוד על באג.
- בהמשך: קובצי הקוד בפרויקט צריכים לכלול הערות לפי הצורך, לכל קובץ לפחות שורת הסבר מה תפקידו.

4. סקר ZFR

במשימה זו עליכם להיפגש עם "נציגי הלקוחות" (צוות הקורס, רישום בעמוד הפגישות) ולהציג להם את מוכנותכם ע"י הצגת:

- רשימת הדרישות הכללית (Product Backlog)
- התשתית או אב הטיפוס מוכנסת למערכת בקרת גרסאות (אין צורך במימוש שום דרישה פונקציונלית).
- דיווח קצר על התכנון לסבב 1 (להלן)

תוצרי הסקר: יש לעדכן את רשימת המשימות בהתאם לדברים שעלו והוחלטו בסקר וכן תיעוד הממצאים בסקר בעמוד הויקי של סבב.

5. תכנון סבב 1

בהגשת סבב 0 תצטרכו כמו בסבבים הבאים גם להגיש תכנון לסבב הבא 1 – (עיינו במשימת הסבבים מספר 6). הסבב הראשון יקרא Minimum Viable Product ובו תדרשו לממש בפעם הראשונה פונקציונליות עיקרית של המערכת שלכם עובדת מקצה לקצה.

6. אב טיפוס

צרו קישור להפצת אב-הטיפוס שלכם. במידת צורך פרטו בעמוד ה-ZFR את תכולות אב הטיפוס.