

פרויקט משימה 5 – פיתוח בסבבים – Iterations

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 דרך השלמת אב הטיפוס ותחילת שימוש במערכת בקרת גרסאות וכן תיעוד ראשוני. (4) תכנון סבב ראשון בשיטת סקראם. בשאר הסבבים תספקו גרסאות הולכות ומתקדמות של המוצר.

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

תוכן

- 1..... מטרת המשימה
- 2..... תאריכים והגשות
- 2..... סבב 0
- 2..... 1. רשימת מאפיינים (סיפורי משתמש) – Product Backlog
- 2..... כללי
- 3..... שלבי העבודה (ראו תמונה)
- 3..... 2. בקרת גרסאות – Version Control
- 4..... 3. תיעוד המוצר
- 4..... 4. סקר ZFR
- 5..... 5. תכנון סבב 1
- 5..... סבבי פיתוח
- 5..... 6. תכנון הסבב הבא
- 6..... 7. מימוש סבב
- 6..... 8. סיום סבב \ רטרופקטיבה
- 6..... 9. מצגת סיום סבב ללקוח
- 7..... 10. משימות סבב ייעודיות

תאריכים והגשות

ההגשה מתבצעת בכל שבוע עד לחצות לפני ההרצאה ע"י שליחת קישורים לעמודי הסבב הרלוונטיים. בכל שבוע ישלח דו"ח התקדמות או סיכום סבב. השליחה דרך טופס המופנה מהויקי. ישנה דרישה שבכל סבב או שבוע יהיה חבר צוות אחר אחראי שיוגדר כ- Scrum Master ויהיה אחראי על כתיבת הסיכום השבועי. בשבוע שבו הגשתם סיכום סבב יש להציג את התוצרים בתרגיל או בהרצאה (תזכורת: לכל צוות יש חובת הרצאה של פעמיים במהלך הסמסטר, לא כולל מצגת הסיום).

ציונים

הניקוד שיינתן לכל סבב יהיה בדגש על הנושאים הבאים:

- איכות המוצר וההתקדמות בפיתוח
- קיום תוצרי הפיתוח המעודכנים (קוד) במאגר הפרויקט
- הצגת הסבב וחווות דעת הלקוח
- ניהול הסבב בויקי, מצב המשימות לאחר הסבב וסיכום – רטרספקטיבה
- מימוש משימות הסבב הייעודיות (ראו סעיף 10)
- תכנון הסבב הבא – משימות ופתיחת דף וויקי

סבב 0

1. רשימת מאפיינים (סיפורי משתמש) – Product Backlog

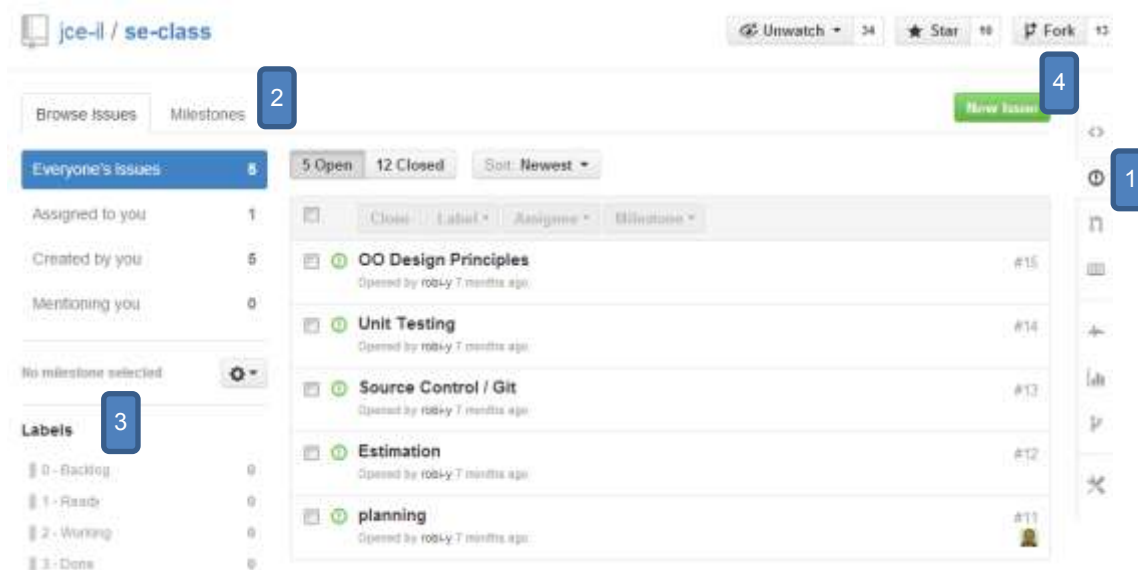


עליכם ליצור טבלת מאפיינים **משויכת, מתועדפת ומוערכת**. המקורות לרשימה זו הם בעיקר סיפורי המשתמש שהכנתם בשלב הדרישות. ישנן מערכות ייעודיות לניהול דרישות ומאפיינים אך אנו נסתפק בהזנת המאפיינים למערכת המשימות של github.



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 מתכנן לתכנן את המשימות ואת מספר הסבבים המתוכנן (בין שניים לארבעה) – כך שתאריך הסיום הוא יום ההרצאה האחרונה בקורס, הסבבים צריכים להיות באורך קבוע)
2. מנהל אתר הפרויקט (המאגר) מכניס במערכת המשימות אבני דרך - אחת לכל סבב מתוכנן (בתפריט הימני באתר: Issues -> כרטיסיית Milestones -> Create a new milestone -> הזנת פרטים: שם משמעותי (למשל Iter0-ZFR), תיאור ותאריך יעד -> Create).
3. בנוסף נעבוד עם לוח ויזואלי שיעזור לכם לראות את מצב המשימות הנוספות. 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). ניתן להוסיף תוויות נוספות לפי הצרכים שלכם (Issues -> labels: שם, צבע -> Create). הוסיפו לפחות תוויות המסמנות את סוג המשימה, למשל: סיפור משתמש/מאפיין, באג, שיפור וכדו'.
4. הצוות נפגש ומחליט על סיפורים ומשימות בפרויקט (לפי המקורות שלעיל עם דגש על תרחישי שימוש וסיפורי משתמש, אך גם תשתיות והכנות נדרשות הידועות כרגע) ולכל משימה מכניסים אותה למערכת (Issues -> New Issue), עם שם, אחראי (Assigned - במידה וכבר ידוע), שיוך לסבב (Milestone) ותוויות. 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 ולאו 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-ספרייה 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 הערות נפרד בויקי).

3. תיעוד המוצר



עליכם ליצור בוויקי הפרויקט דפי תיעוד למשתמש ולמפתח. 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 כלליים על המוצר
- כיצד להתקין 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 (readme.md) נועד עבור מפתחים חדשים שמצטרפים אליכם לפיתוח, לבדוק אותו או להשתמש בו למוצר אחר. הדף למפתח יהיה על בסיס הדפים שכבר הכנתם לתיאור המוצר ותהליך הפיתוח – אך יש להעביר את תוכנם אל הקובץ README.md שבספריית השורש שלכם, כך שהוא יוצג כעמוד הבית של הפרויקט שלכם (או לפחות לקשר אל המידע בויקי באופן ברור מעמוד הבית).

הוא צריך לכלול:

- כיצד להשיג את הקוד (גרסה אחרונה וגם כתובת המאגר)
- הערה בדבר רישיון הפרויקט (למשל קוד פתוח BSD עם זכויות מסוימות למכללה)
- פרטים 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.
- קובצי הקוד צריכים לכלול הערות לפי הצורך, לכל קובץ לפחות שורת הסבר מה תפקידו.

4. סקר ZFR

במשימה זו עליכם להיפגש עם "נציגי הלקוחות" (צוות הקורס) 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). כחלק מהשלד הזה יהיה גם קובץ התיעוד Readme.md שלעיל
- דיווח קצר על התכנון לסבב 1 (להלן)

תוצרי הסקר: יש לעדכן את רשימת המשימות בהתאם לדברים שעלו והוחלטו בסקר וכן תיעוד תוצרי הסקר בעמוד הוויקי של סבב 0 (ראו בהמשך).

5. תכנון סבב 1

בהגשת סבב 0 תצטרכו כמו בסבבים הבאים גם להגיש תכנון לסבב 1 – ראו סעיפים 6 ו-10 להלן. הסבב הראשון יקרא Minimum Viable Product ובו תדרשו לממש בפעם הראשונה פונקציונליות עיקרית של המערכת שלכם עובדת מקצה לקצה.

סבבי פיתוח

6. תכנון הסבב הבא



במקביל לסיום העבודה על הסבב הנוכחי, יוגש גם תכנון עבור הסבב המתקרב (n+1). התכנון כולל בחירת סיפורים ותכנון משימות בתיאום עם הלקוח. פתיחת עמוד וויקי עבור הסבב ועדכון ה-product backlog וה-sprint backlog.

תכנון משימות: בתיאום עם הלקוח וצוות הקורס נפגשים ובוחרים את הסיפורים והמאפיינים, בעלי הערך הגבוה ביותר כרגע, למימוש (למשל עבור ארבעה סבבים - כרבע מכלל הסיפורים). הסיפורים מועברים מה-product backlog אל ה-sprint backlog ע"י שיוך לסבב (אבן דרך ב-github issues) וע"י גרירה אל העמודה Ready בלוח huboard וגרירה למיון לפי עדיפות המשימה.

אפשר גם להוסיף משימות שכבר ידועות מראש על פי תוצרי השלבים הקודמים (כגון: רעיון (SOW), אתחול (SDP), SRS ו-SDS), משימות הפרויקט המוגדרות כאן ובפרט בסעיף 10 וכל מה שלדעתכם מצריך עבודה של הצוות. ישנן מערכות ייעודיות לניהול משימות אך אנו נסתפק ב-github issues הפשוטה. במיוחד כל סיפור שנבחר, מפרטים לתת-משימות פיתוח לפי המימוש הנדרש. כל תת-משימה כזו מוכנסת למערכת כולל הערכת המאמץ הנדרש בשעות, תיוג, שיוך וכדו'. לפי הצורך מעדכנים את הערכות המאמץ והזמנים לפי הניסיון שנצבר.

בדיקות: בטקסט של הסיפורים העיקריים לסבב זה יש לפרט תרחיש בדיקה (קבלה) שמדגים כיצד תוודאו שהסיפור הושלם.

מומלץ: ניהול תת-המשימות של סבב 0 גם כן במערכת זו. בבחירת המשימות יש להתייחס לערך המקסימלי שהלקוח יכול לקבל מעבודתכם ומצד שני לגודל המשימות בהתאם להערכת הזמנים שלכם.

בוויקי יש להוסיף עמוד חדש עבור הסבב הקרוב (ולקשר אותו לעמוד הראשי readme). העמוד יהווה לוח עבודה משותף להעלאת נושאים בין חברי הצוות במשך העבודה בסבב. העמוד יכלול לפחות:

- כותרת עם שם הסבב
- מטרות הסבב
- בעלי תפקידים מהצוות בסבב זה (Product Owner, Scrum Master)
- מידע על מצב המשימות, לפחות בצורה של קישור לטבלאות משימות פתוחות וסגורות לסבב זה וכן ללוח ויזואלי של המשימות, כגון:

<https://github.com/jce-il/se-class/issues?milestone=1&state=open>

<https://huboard.com/jce-il/se-class/backlog>

- מספר נקודות ההערכה מתוכננות למימוש (ע"י סיכום הנקודות של המשימות השונות)
- הפניות לתרשימים נדרשים לצורך המימוש בסבב למשל class diagram או כרטיסי CRC.
- סיכומי פגישות, חוות דעת של הלקוח וסיכומים שבועיים – בפרט יש לגבות את החלטותיכם לגבי

המאפיינים שנבחרו והיקפם.

- סיכום הסבב (ימולא בסיומו; ראו פירוט להלן בסעיף 8)
- כל מידע אחר שדרוש לכם לצורך שיתוף פעולה בסבב

7. מימוש סבב

תוך כדי המימוש המשימות מועברות עמודות בלוח huboard ונסגרות בסיומן (Ready <- Working - Close, Done <).

כאמור, עבור הסיפור העיקרי הממומש בסבב יש להגדיר תרחישי בדיקה שיובילו את המימוש. מאגר הקוד מתעדכן באופן רציף בתוצרי עבודתכם.

8. סיכום סבב \ רטרוספקטיבה



עמוד הויקי של הסבב יכול לז"ח (או קישור למסמך) המסכם ישיבת רטרוספקטיבה שביצעתם לסיכום הסבב ובה בין השאר הנושאים הבאים:

- תכנון מול ביצוע - סיכום של נקודות הערכה שבוצעו והשוואה לסבבים קודמים (velocity).
- מה עבד טוב ומה פחות בסבב? למשל:
 - מהו מצבו הנוכחי של המוצר, האם עמדנו בדרישות? האם הוא איכותי מספיק?
 - האם הבדיקות שנכתבו מראש עזרו?
 - האם השיטות שהשתמשנו בהן לפיתוח עד כה היו טובות? מה ניתן לשפר לפעם הבאה?
 - כיצד עבדה הקבוצה עד עכשיו? האם כולם תרמו את חלקם? מהם אופני ותדירות התיאומים בינכם? מה לא עבד טוב? האם וכיצד ניתן לשפר? מה הוחלט לשנות באופן העבודה?
 - אפשר גם לציין לקחים הקשורים לחומרי הקורס ההרצאה והתרגיל והאם עזרו (או להיפך..) להשלמת הסבב
- כיצד מתקדם הפרויקט? האם יעמוד בל"ז ובמשימות? כמה נקודות הערכה הספקנו לבצע? אלו סיכונים עדיין קיימים? מהי תכנית ההתמודדות איתם?

חובה לכלול חוות דעת שהתקבלה מהלקוח על תוצרי הסבב.

יש לסגור את המשימות והסיפורים שהסתיימו.

יש להוסיף בדף הסבב קישור לגרסת הרצה של המוצר.

9. מצגת סיום סבב ללקוח

לקראת סיום העבודה על הסבב אתם מתבקשים להציג את מצב המוצר (ולחירשם להצגה בעמוד הפגישות). ללקוח עצמו הכי טוב להראות את מצב התוכנה, אך עבור כאלו שלא מצליחים להגיע לפגישות (הבודק?) הטוב והפשוט ביותר הוא להוסיף מספר לכידות מסך של עבודה עם המוצר והסברים לפי הצורך.

מה נדרש? הדגמה קצרצרה! של עבודה עם המוצר! יש להיערך **מראש** עם כל ההתקנות והחיבורים הדרושים ופשוט למשוך את הקוד מהמאגר ולהדגים את הפונקציונליות העיקרי שמושה. אם יש דברים שלא הספקתם לסיים (Done) ציינו זאת בקצרה (לא צריך להתבייש...)

אם יש זמן אפשר גם להציג לקחים מעבודתכם עד כה (ראו סעיף הבא).

לסיום הציגו בקצרה את מצב המשימות והתכנון לסבב הבא.

רשימת המשימות צריכה להיות מעודכנת כך שהמשימות של סבב זה במצב Done, למעט משימות שעוברות לסבבים הבאים.

10. משימות סבב ייעודיות

הערה: המשימות עלולות להתעדכן בהתאם לחומר הנלמד ובמקרה כזה יפורסם עדכון. אם לא צוין מספר סבב אתם קובעים באיזה סבב לבצע משימות אלו – מומלץ בסבב העוקב להרצאה שבה נלמד החומר הרלוונטי

1. MVP (סבב 1)

בסבב זה יש לממש פונקציונליות עיקרית של המערכת שלכם עובדת מקצה לקצה. בפרט יש ליצור חלקים מהשכבות השונות הנדרשות להדגים את יכולתו העיקרית של המוצר. ייתכן מאד ששכבות אלו מאד ראשוניות – למשל ממשק משתמש עם פונקציונליות בסיסית, בסיס-נתונים דמה להחזרת תשובות קבועות וכדו'. בנוסף יש להתייחס לסיכון העיקרי בפרויקט על ידי טיפול בו ודיווח על סטטוס במצגת סיום הסבב.

2. בדיקות יחידה (סבב 2)

עליכם לבחור מחלקה עיקרית שיש לממש בסבב זה. יש לכתוב עבורה סט בדיקות משמעותי – לפני המימוש. מומלץ לבחור מחלקה מרכזית אך בעלת תלות נמוכה ברכיבים אחרים. את הבדיקות יש להציג בסקר בדיקות (בהרצאה או בתרגיל) שיתקיים שבוע אחד לפני סוף הסבב. המימוש יתבצע עד לסוף הסבב. בפגישת סוף הסבב דגש על קוד מעודכן במאגר. במערכת המשימות תהיה משימה של בדיקות יחידה סגורה עם קישור לקוד הבדיקות.

3. סקר שיפורי קוד – Refactoring (לבחירתכם, לא בסבב האחרון)

בצעו במהלך הסבב סקר פנימי או עם נציג מצוות הקורס לגבי איכות הקוד הנוכחי ועמידתו לשינויים עתידיים. מהלך הסקר:

- הציעו רעיון להרחבת המוצר שלא נלקח בחשבון בתיכון עד כה.
- זהו את המקומות בקוד שבהם יש לבצע את השינוי או ההרחבה.
- זהו 3-5 בעיות בתיכון קוד זה או "ריחות" שיקשו על השינוי המתבקש.

יש להפנות לבעיות מרשימה זו:

<http://users.csc.calpoly.edu/~jdalbey/305/Lectures/SmellsToRefactorings>

- תארו באלו שיפורי קוד (Refactorings) מתוך הרשימה תשתמשו כדי לתקן את הקוד (תוכלו להשתמש בהצעות אלו למשימת הסבב הבא).

הגשה – א. קישור לסקר מדף הסבב בוויקי ובמייל הגשת הסבב. ב. התייחסות קצרה לנושאים שעלו במצגת סיום סבב זה.

4. סקר קוד (לבחירתכם)

עליכם לבצע סקר קוד דרך מערכת ה-pull request ב-github או באמצעות הערות על commit מסוים. על כל חבר צוות לבצע לפחות סקר אחד. מומלץ להשתמש כקלט בתוצרי הסקר מהסבב הקודם. ניתן להתייחס גם לנושאי חווית משתמש.

הגשה: בדף סבב הסיום תהיה פסקה של סקרי קוד. הפסקה תכיל לכל חבר צוות קישור ל-issue (א) (pull request) שנפתח עבור סקר קוד שבוצע על קוד שהוא כתב. ה-issue יכיל את הדו-שיח בין הנסקר לסוקר וקישורים מתאימים למאגר ה-pull, diff או הערות הקשורות לסקר. (המלצות: שימוש בסקר שיפורי הקוד מהסבב הקודם, הוספת בדיקות כיסוי כשלב מקדים לשינוי, התייחסות בשינויים לחומרי הקורס האחרונים – תיכון, תבניות, UX-i refactoring)

5. סיכום פרויקט (סבב אחרון)

1. מילוי סקר (אנונימי) להערכת השיטות, הכלים והתהליכים שלמדנתם בקורס. התוצאות יוצגו בהרצאה האחרונה בקורס (קישור יפורסם).
2. מצגת סופית – בהרצאה האחרונה בסמסטר עליכם להציג א. את המוצר הסופי ב. לקחים עיקריים מהפרויקט. כל חברי הצוות נדרשים להציג וייתכנו שאלות (ניקוד אישי). יש להירשם בעמוד הפגישות.
3. במסגרת סיכום הסבב כרגיל בצעו גם סיכום מפורט יותר לפרויקט כולו. עליכם להגיש דו"ח הכולל תקצירים וקישורים עבוד הנקודות הבאות:
 - א. סיכום הסבב האחרון, בדומה לקודמים.
 - ב. עדכון מסמכי תיעוד המוצר, למשל בעמידות המוצר לתקלות שונות: קלטים שגויים, שינויים בסביבת ההתקנה וההפעלה (ניתוקי רשת) וכדו'.
 - ג. דיון על רחלערכי הצוות. הערכת המידה בה הצוות עבד לפי רשימת "רוח הצוות" שפורסמה באתר בתחילת הדרך. תנו דוגמאות ספציפיות מניסיונכם על מנת לאשש את המסקנות.
 - ד. רפלקציה על הפרויקט. רשמו את שלשת הלקחים העיקריים שלמדנתם מהפרויקט שליווה את הקורס. הלקחים יוכלו להיות לפי שלבי הפרויקט השונים (הצעה, מפרט דרישות, תכנון, בדיקות, מימוש, עבודת צוות וכדו'), מה עבד טוב? מה פחות? מה תעשו אחרת בפרויקט הבא? עם אילו אתגרים וסיכונים התמודדתם, האם הרצאת האורח נתנה פרספקטיבה שונה? אלו כלים סייעו ואלו פחות? האם העבודה מול הלקוח הייתה משמעותית ותורמת? האם יש לכם תכניות להמשיך את פיתוח המוצר/להעביר אותו לקוד פתוח וכדו'?
 - ה. הערות כלליות על הקורס