



## פרויקט 5 – פיתוח בסבבים – Iterations

### תוכן

1.....	מטרת המשימה	1
2.....	תאריכים והגשות	2
2.....	הקדמה: כמה מילים על סקראם (שיטת ניהול פרויקטים)	2
3.....	סבב 0	3
3.....	1. רשימת מאפיינים (סיפורי משתמש) – Product Backlog	3
3.....	כללי	3
4.....	שלבי העבודה	4
4.....	2. בקרת גרסאות – Version Control	4
4.....	3. תיעוד המוצר	4
5.....	4. סקר ZFR	5
5.....	5. תכנון סבב 1	5
5.....	סבבים 1-4	5
5.....	6. תכנון סבב	5
6.....	7. מימוש סבב	6
6.....	8. מצגת סיום סבב ללקוח	6
6.....	9. סיכום סבב	6
7.....	10. משימות סבב ייעודיות	7

### מטרת המשימה

חברת סבב"ה מקפידה על פיתוח איכותי ולכן דורשת ניהול מסודר ומונחה מטרה וכן שימוש בכלי הנדסת תוכנה לניהול הפרויקט ובקרת גרסאות הקוד. בנוסף הוחלט שהפיתוח המתאים לפרויקט הוא פיתוח בסבבים. המשימות הבאות מחולקות לסבב 0 שבו תכינו את התשתיות לעוד ארבעה סבבי פיתוח עד לסיום הקורס. סבב 0 נקרא גם לפעמים Zero Feature Release מכיוון שהוא מתמקד בהתארגנות ניהולית ותשתיתית. 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) הכנת כלי הפיתוח הנדרשים ובמיוחד סביבת הפיתוח ומערכת בקרת גרסאות, כולל מימוש תשתית למוצר ותיעוד ראשוני. 4) תכנון סבב ראשון (ספרינט \ איטרציה) בשיטת סקראם. בשאר הסבבים תספקו גרסאות הולכות ומתקדמות של המוצר.

לניהול המשימות נשתמש במערכת github/issues ובלוח [gissues](#) המבוסס עליה. הקוד יאוחסן כמובן ב-github ע"י שימוש ב-git.

## תאריכים והגשות

להרצאה הבאה 29/4 - הגשת תוצרי סבב Iter0-ZFR 0 (ההגשה מתבצעת כרגיל ע"י שליחת קישור למייל הקורס לדף הסבב כמפורט בהמשך ובמידת הצורך הפניות והסברים נוספים)  
להרצאה ולתרגיל הבא - הצגת גרסת ZFR בכיתה (כ-10 דקות)

לסבבים הבאים עד ליום ההרצאה/תרגיל יש להכין את סיכום ומצגת הסבב הנוכחי ואת תכנית הסבב הבא. בכל סבב תינתן משימת משנה עם הוראות משלה לפי ההתקדמות בחומרי ההרצאה (סעיף אחרון במסמך זה). כמו כן לעיתים יתקיים סקר מוקדם על משימה ספציפית לסבב (ועל כל קבוצה להציג לפחות פעם אחת בהרצאה)


הגשת סבב 1 Iter1-MVP – 13/5  
הגשת סבב 2 Iter2-TDD – 27/5 (+סקר בדיקות)  
הגשת סבב 3 Iter3-Refactoring – 10/6 (סקר תיכון – 3/6)  
הגשת סבב 4 Iter4-CodeReview – 24/6 (סקר קוד – 17/6)

## צינונים

הניקוד לכל סבב יהיה לפי הקטגוריות הבאות:

- איכות המוצר בדגש על ההתקדמות בפיתוח – 20%
- העלאת התוצרים המעודכנים למאגר הפרויקט – 5%
- הצגת הסבב וחווית דעת הלקוח – 10%
- מצב המשימות לאחר הסבב וסיכום העבודה – רטרספקטיבה – 15%
- המשימה המיוחדת לסבב (ראו סוף המסמך) – 40%
- תכנון הסבב הבא – משימות ופתיחת דף וויקי – 10%

## הקדמה: כמה מילים על סקראם (שיטת ניהול פרויקטים)

להלן רקע מקדים לשיטת עבודה דומה לנדרש מעתה, נרחיב בהרצאת המשך. מובא מתוך   
בלוג "אג'יל זו לא מילה גסה" של אלעד סופר - יועץ בנושא: [blog.thescrumster.com/archives/25](https://blog.thescrumster.com/archives/25)  
גרסה עדכנית יותר: [blog.thescrumster.com/?p=25](https://blog.thescrumster.com/?p=25)

[בסקראם] ישנו אחראי מוצר שמייצג את הלקוח (Product owner) שאחראי לתחזוקה של רשימת הדרישות הממוינת לפי ערך ללקוח (Product backlog), בכל תחילה של ספרינט (מיד תבינו בדיוק מה זה ספרינט), בישיבת תכנון הספרינט, הצוות בוחר לפי סדר העדיפות שברשימה על איזה פריטים הוא יכול להתחייב שיפותחו בספרינט הקרוב, ובסוף הספרינט הצוות מציג את המוצר עם התוספות שהסתיימו (DONE) בספרינט לכל מי שמעוניין. מה זה DONE? עוד רגע...

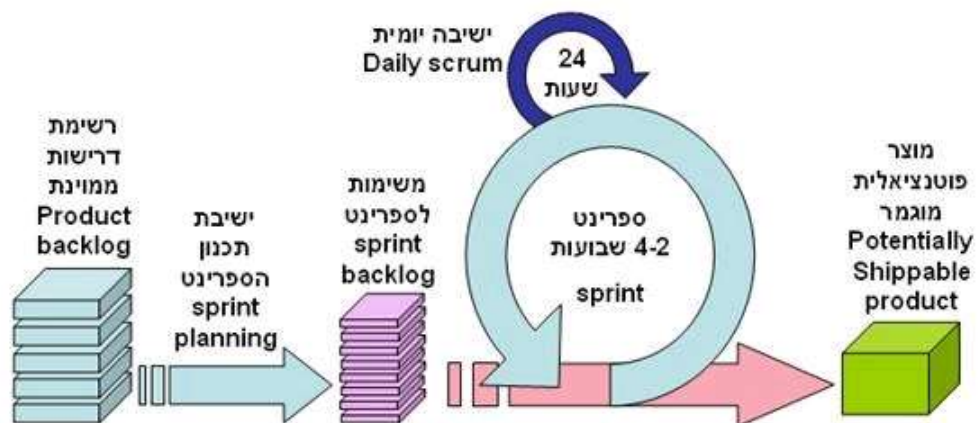
מהו הספרינט? תקופה שבין 15-30 ימים שבה הצוות עובד על פיתוח הפריטים שנבחרו בתחילת הספרינט. במהלך הספרינט אין להפריע לצוות, אין לשנות עדיפות, ואין לשנות תכולה. בכל יום במהלך הספרינט ישנה ישיבה (Daily) של עד 15 דקות, שבה חברי הצוות מעדכנים זה את זה במצב המשימות ע"י מענה על שלוש שאלות פשוטות: מה עשיתי עד עכשיו? מה אני אעשה עד הישיבה הבאה? ומה מפריע לי? את הישיבה הזאת, כמו גם את תכנון הספרינט מוביל הסקראם מסטר (Scrum master). תפקידיו של הסקראם מסטר הם פשוטים, קשים וחשובים גם יחד: לדאוג שהסקראם יתנהל כמו שצריך, לדאוג שלא יפריעו לצוות לעבוד, לדאוג להסיר מכשולים שיש לצוות, ולהנחות את הישיבות של הסקראם. הסקראם מסטר אינו מוביל הצוות, הוא מוביל הסקראם, הצוות הוא יחידה עצמאית לגמרי.

בכל סוף ספרינט מציגים לכל מי שמעוניין את התוספת למוצר אשר פותחה מתוך הפריטים ואשר הם DONE. מה זה DONE ? DONE זה הקריטריון שמוסכם על הצוות ועל מנהל המוצר אשר מגדיר מתי פריט מהרשימה נחשב גמור, למשל DONE = Implemented, unit tested, acc tested.

בנוסף לכך בסוף כל ספרינט מתקיימת ישיבת רטרוספקטיבה, ישיבה זו נועדה לשקף את הספרינט האחרון ולדון מה היה טוב ויש לשמר ומה היה פחות מוצלח וניתן לשפר.

גם הצוות וגם מנהל המוצר מתחזקים גרף שנקרא Burn down chart, גרף זה מספק תמונה טובה על התקדמות הצוות/המוצר בספרינט \ בפרויקט.

בכל זמן נתון אפשר לשנות את רשימת הדרישות ולמיין אותה אחרת, והצוות מתחייב בכל פעם רק על ספרינט אחד. קבלו דיאגרמה פרי עטי (יצא לי טוב דוקא).



## סבב 0

### 1. רשימת מאפיינים (סיפורי משתמש) – Product Backlog



מכיוון שהתנהלות הפרויקט מתוכננת להיות בשיטת סקראם (נלמד יותר בהמשך) עליכם ליצור טבלת מאפיינים **משיכות, מתועדפת ומוערכת**. המקורות לרשימה זו הם בעיקר סיפורי המשתמש שהכנתם בשלב הדרישות. אפשר גם להוסיף משימות שכבר ידועות מראש על פי תוצרי השלבים הקודמים, כגון: הצעה (SOW), אתחול (SDP), SRS ו-SDS, משימות הפרויקט השונות הניתנות בקורס וכל מה שלדעתכם מצריך עבודה של הצוות. ישנן מערכות ייעודיות לניהול משימות אך אנו נסתפק ב-github issues הפשוטה.




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 (למשל רק עם נקודות-סיפור). לפני כל סבב מנסים לחלק למשימות קטנות יותר עד לרמה שעבורה כבר אפשר לתת הערכה גסה לגבי יחסיות הזמן הנדרש לכל משימה. מתעדים את ההערכות שניתנו ומזינים זמני עבודה כדי שבהמשך נוכל לשפר אותן על סמך הניסיון שנרכש בהערכה.

## שלב העבודה

- מנהל אתר הפרויקט (המאגר) מכניס חמישה סבבים (ב-github נקראים אבני דרך), אחד לכל סבב מתוכנן (Issues <- כרטיסיית Milestones <- Create a new milestone <- הזנת פרטים: שם (למשל Iter0-ZFR), תיאור ותאריך יעד <- Create).
  - רשות: תוויות: ניתן להכין תוויות כלליות (labels) לסימון המשימות השונות (Issues <- labels: שם, צבע <- Create). תוכלו להכין תוויות לפי הצרכים שלכם או להשתמש בקיימות.
  - הצוות נפגש ומחליט על משימות 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) וסבב (Milestone).
- 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 ואו 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 קבצי ניסיון טכנולוגי שביצעתם (במילים פשוטות: להעלות מספר קבצי קוד ל-github)

הצעדים הנדרשים:

1. יצירת פרויקט בסביבת העבודה שלכם והוספתו למערכת בקרת הקוד - git
2. תשתיות שהכנתם מוספים גם כן
3. אופציה: הוספת תת-ספרייה עבור מסמכי הפרויקט (SDF – Software Document Folder)
4. דחיפת הקוד למאגר של הפרויקט ב-github

## 3. תיעוד המוצר

 עליכם ליצור בוויקי הפרויקט דפי תיעוד למשתמש ולמפתח כולל דפי הורדות (או קישורים) לגרסה בינארית וגרסת קוד. הדף למפתח יהיה על בסיס הקובץ README.md שבספריית השורש שלכם, כך שהוא יוצג כעמוד הבית של הפרויקט שלכם (בדומה לאתר הקורס). 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 להוריד, להתקין ולהריץ את המוצר
- פרטים ראשוניים על אופן השימוש במוצר.
- כיצד לדווח על באג שמתגלה (למשך כתובת מייל או פתיחת נושא במערכת המשימות וכדו')

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

התיעוד למפתח (readme.md) נועד עבור מפתחים שרוצים להצטרף אליכם לפיתוח, לבדוק אותו או להשתמש בו למוצר אחר. 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 המקור צריכים לכלול הערות לפי הצורך, לכל קובץ לפחות שורת הסבר מה תפקידו.
- קישורים לתיעוד המוצר ועמוד הורדות – אפשר להכין עמוד וויקי נפרד או פשוט להפנות לעמוד ההורדות של github.

#### 4. סקר ZFR

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

- רשימת הדרישות הכללית (Product Backlog)
- התשתיות או הארכיטקטורה שהספקתם להכין או אב טיפוס מעל תשתיות קיימות, כך שהלקוח ישתכנע שעכשיו תוכלו להתחיל לממש פונקציונליות.
- תצטרכו להראות שהקוד מוכנס למערכת בקרת גרסאות ושהוא מכיל שלד שמתקמפל ורץ בסביבת עבודה – אך ייתכן שעדיין אינו מספק שום דרישה פונקציונלית. כחלק מהשלד הזה יהיה גם הקובץ התיעוד Readme.md
- 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 הסבב (ראו בהמשך).

#### 5. תכנון סבב 1

בהגשת סבב 0 תצטרכו גם להגיש תכנון לסבב 1 כמפורט בסעיף הבא. הסבב הראשון יקרא Minimum Viable Product ובו תדרשו לממש בפעם הראשונה פונקציונליות עיקרית של המערכת שלכם עובדת מקצה לקצה (פרטים נוספים יימסרו לקראת ההצגה של סבב זה).

### סבבים 1-4

#### 6. תכנון סבב

לכל סבב מתקרב יש לבצע במקביל לסיום העבודה על הסבב תכנון שיוגש יחד עם סיום הסבב הנוכחי. 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 בוחרים סיפורים ומאפיינים למימוש (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 not started בלוח gissues וגרירה למיון לפי עדיפות המשימה. כל סיפור מפרטים למשימות משנה לפי המימוש הנדרש. כל תת-משימה כזו מוכנסת למערכת כמו הקודמות כולל הערכת המאמץ הנדרש, תיוג, שיוך וכדו'. לפי הצורך מעדכנים את הערכות המאמץ והזמנים לפי הניסיון שנצבר.

בדיקות: בטקסט של הסיפורים העיקריים לסבב זה יש לפרט תרחיש בדיקה שמדגים כיצד לוודא שהסיפור הושלם.

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

בוויקי יש להוסיף עמוד חדש עבור הסבב הקרוב. העמוד יהווה לוח עבודה משותף להעלאת נושאים בין חברי הצוות במשך העבודה בסבב. העמוד יכלול לפחות:

- כותרת עם שם הסבב

- מטרות הסבב

- 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>

<http://gissues.com/?repo=jce-il%2Fse-class>

- מספר נקודות ההערכה מתוכננות למימוש

- תרשימים נדרשים לצורך המימוש בסבב למשל class diagram או כרטיסי CRC.

- סיכומי פגישות וחווות דעת של הלקוח – בפרט יש לגבות את החלטותיכם לגבי המאפיינים שנבחרו והיקפם.

- סיכום הסבב (ימולא בסיומו; ראו להלן)

- כל מידע אחר שדרוש לכם לצורך שיתוף פעולה בסבב

## 7. מימוש סבב

תוך כדי המימוש המשימות מועברות עמודות בלוח gissues ונסגרות בסיומן.

תרחישי הבדיקה שבסיפורים העיקריים מבוצעים לוודא נכונות.

מאגר הקוד מתעדכן באופן רציף בתוצרי עבודתכם.

## 8. מצגת סיום סבב ללקוח

לקראת סיום העבודה על הסבב אתם מתבקשים להציג את מצב המוצר (ולהירשם 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 לא מתאפשר אפשר להכין מצגת קצרה. אם יש זמן אפשר גם להציג לקחים מעבודתכם עד כה.

לסיום הציגו בקצרה את מצב המשימות והתכנון לסבב הבא.

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

## 9. סיכום סבב

עמוד הוויקי של הסבב יכלול דו"ח (או קישור למסמך) המסכם ישיבת רטרוספקטיבה שביצעתם לסיכום הסבב ובה בין השאר הנושאים הבאים:

- 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 לציין בעמוד הוויקי סיכום של נקודות הערכה שבוצעו והשוואה לסבבים קודמים (velocity). במידה ולא הספקתם לבצע ישיבת סיכום עד למצגת הלקוח – ניתן להגיש חלק זה באיחור של יום נוסף.
- חובה לכלול חוות דעת שהתקבלה מהלקוח על תוצרי הסבב.

### 10. משימות סבב ייעודיות

הערה: המשימות עלולות להתעדכן בהתאם לחומר הנלמד ובמקרה כזה יפורסם עדכון

#### סבב 1 – MVP

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.

#### סבב 2 – בדיקות

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 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.

#### סבב 3 – תיכון מונחה עצמים (UX+)

שלב א - בצעו במהלך הסבב סקר עם לקוח טכני או נציג מצוות הקורס (יש לתאם פגישה) לגבי איכות התיכון הנוכחי ועמידותו לשינויים עתידיים. יש להתחשב בעקרונות תיכון שנלמדו בכיתה או אחרים. תעדו את הנושאים שעלו ובמיוחד זהו רכיב המתאים להרחבה עתידית ומהם השיפורים הנדרשים להרחבה זו – בסבב האחרון נשתמש גם בלקחי סקר זה לשיפור הקוד.

שלב ב – הצגת הקוד שלכם בכיתה במסגרת סקר תיכון בהרצאה והתרגיל – יש להתייחס לנושאים שעלו בסקר שיפורי התיכון עם הלקוח.

#### סבב 4 – סקר קוד \ CI \ UX וסיכום הפרויקט

1. סקר קוד – 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.

הגשה: בדף סבב 4 תהיה פסקה של סקרי קוד. הפסקה תכיל לכל חבר צוות קישור ל-issue (או pull request) שנפתח עבור סקר קוד שבוצע על קוד שהוא כתב. ה-issue יכיל את הדו-שיח בין הנסר

- לסוקר וקישורים מתאימים למאגר ה-pull, diff או הערות הקשורות לסקר.  
(המלצה: לבצע שינויים הקושרים לחומרי הקורס האחרונים – תבניות, UX-i refactoring)
2. בונס 10 נק' – הפצה מתמשכת. הגדירו תהליך הפצה מתמשכת בו ניתן ע"י דחיפה של קוד למאגר להתחיל תהליך אוטומטי של הידור, בדיקות והפצת האפליקציה. חובת הצגה בהרצאה האחרונה של התהליך.
3. מצגת סופית – בהרצאה עליכם להציג את המוצר הסופי וכן לקחים עיקריים מהפרויקט. כל חברי הצוות נדרשים להציג וייתכנו שאלות.
4. 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 שלמדתם 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
  - ו. ~~הערכת עמיתים (לא תבצע בקורס זה)~~