

מפרט אבן דרך 0 - Milestone Zero

Zero Feature Release

תוכן

1	מטרת המשימה
1	תאריכים
2	הקדמה: כמה מילים על סקראם (שיטת ניהול פרויקטים)
4	1.רשימת משימות – Product Backlog
	כללי 4
4	שלבי העבודה
4	2.בקרת קוד – Source Configuration Management
4	3.סקר ZFR
5	תיעוד המוצר
5	4.תכנון אבן דרך מס' 1
6	נספח – צעדים ליצירת פרויקט והכנסתו למערכת בקרת קוד Mercurial בסביבת מיקרוסופט

מטרת המשימה

החברה שהקמנו עבור פיתוח זיהוי פרוסות סיליקון חריגות מקפידה על פיתוח איכותי ולכן דורשת ניהול מסודר ומונחה מטרה וכן שימוש בכלי ניהול ובקרת קוד. משימה זו מוגדרת כאבן דרך המוגשת לחברה. הפעם כבר אין מסמכים להגשה, אלא תחילת עבודה בא. מטרת משימה זו היא להכין את מירב התשתיות הדרושות למימוש דרישות המוצר. המשימה כוללת:

- 1) הכנת רשימת משימות ממוינת לפי צרכי הלקוח ומחולקת לפי איטרציות ושחרורים.
- 2) הכנת כלי הפיתוח הנדרשים ובמיוחד מערכת בקרת הקוד.
- 3) בשלב זה יתבצע כבר מימוש תשתיות נדרשות (הכוונה לכתיבת קוד), כך שבסופו יתבצע שחרור ZFR (Zero Feature Release) כולל תיעוד ראשוני.
- 4) תכנון ספרינט ראשון בשיטת סקראם.

תאריכים

שיקלו כיצד לנהל את גם את שלל המשימות בתרגיל זה ובאיזה לוחות זמנים. לדוגמא

- 10:00 22/11 - הגשה במייל (exs.ice@gmail.com) של קישור לרשימת הדרישות שב-FogBugz וכן העתק מיוצא לפורמט CSV.
- 24:00 24/11 - הגשה במייל של העתק דו"ח על הכנסת מסמכי פרויקט וכן חליפת בדיקות יחידה ראשונה למערכת בקרת הגרסאות Kiln.
- 10:00 29/11 - הגשה במייל של עותק גראף Burn down לאבן דרך 0 וקישור לדף עם חומרי אבן הדרך.
- 12:00 29/11 - הצגת גרסת ZFR ללקוחות בזמן התרגיל (הציון גם על איכות הגרסה).
- פגישת תכנון עם נציגי הלקוח לאיטרציה ראשונה.
- 24:00 29/11 - הלקוחות מזינים את הערותיהם ב-FogBugs עם עותק לבודק.
- הגשת קישור לדפים עם סיכום פגישת התכנון לאיטרציה ראשונה + sprint log.
- מילוי שאלון הערכת עמיתים בכתובת.

בסיום משימה זו במערכת הניהול, ימצאו:

- רשימת המשימות במצבה הנוכחי עם המשימות השונות מתועדפות.
- עותקים של מסמכי הפרויקט מעודכנים.

הקדמה: כמה מילים על סקראם (שיטת ניהול פרויקטים)

מקור:  <http://blog.thescrumster.com/archives/25>

בסקראם ישנו **אחראי מוצר (Product owner)** "שמייצג" את הלקוח, הוא אחראי 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 ?

מה מפריע לי ?

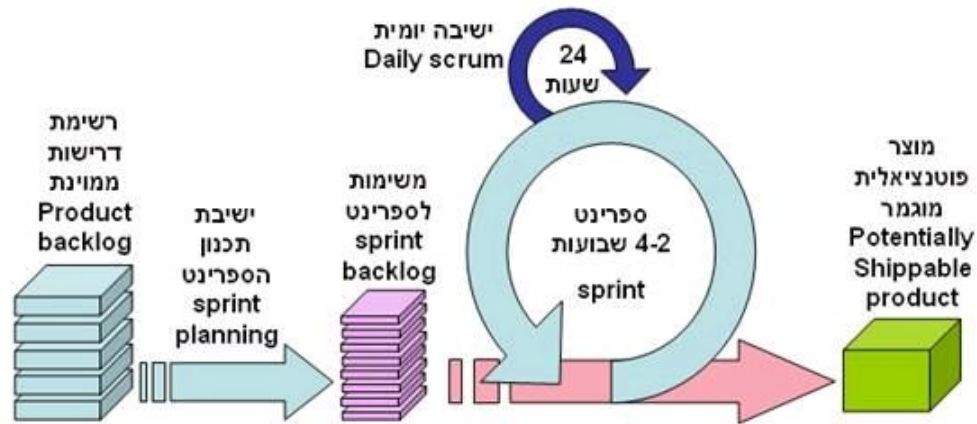
את הישיבה הזאת, כמו גם את תכנון הספרינט מוביל ה**סקראם מסטר (Scrum master)**. תפקידיו של הסקראם מסטר הם פשוטים, קשים וחשובים גם יחד: לדאוג שהסקראם יתנהל כמו שצריך, לדאוג שלא יפריעו לצוות לעבוד, לדאוג להסיר מכשולים שיש לצוות, ולהנחות את הישיבות של הסקראם. הסקראם מסטר אינו מוביל הצוות, הוא מוביל ה**סקראם**, 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 ולמייין אותה אחרת, והצוות מתחייב בכל פעם רק על ספרינט אחד. קבלו דיאגרמה פרי עטי (יצא לי טוב דוקא).



1. רשימת משימות - Product Backlog

7 שבועות עד לסיום הסמסטר <-- בערך יהיו 3 איטרציות

רשימת הנושאים שצריכים להיות מטופלים במסגרת המוצר המפותח:

צריך לקבוע עם פרויקט 1 בין 2 האופציות הבאות: את קובץ הטקסט (שמהווה קלט עבורנו) הם שומרות במסד הנתונים ואנחנו נפנה לשם כאשר נקבל עדכון לגבי זה שיש קלט למערכת, או שבשביל שהכל יהיה רציף לגמרי, שלא ישמרו את קובץ הטקסט אלא יעבירו לנו אותו ישירות...

באיטרציה 1:

- יש ללמוד את סביבת העבודה, השפה ++c בסביבת העבודה visual studio, תהליך הלמידה כולל הורדה והתקנה.
- גישה לקלט מפרויקט 1.
- פתיחת הקובץ לקריאה.
- יצירת משתנה חדש (מחרוזת) דינאמי שיכיל את המידע שקוראים מהקובץ.
- קריאת הקובץ עד ל EOF.
- פתיחת קובץ חדש שהוא יהיה הפלט שלנו- שנכתוב אליו את הסדרה האלפביתית.
- יצירת רשימה מקושרת שתכיל בתוכה תת מחרוזת עם התאמה לערך מעל הא"ב, הרשימה תהיה בגודל דינאמי.
- חלוקת המחרוזת המקודדת הבינארית לתת-מחרוזות הנקראות פסקאות בתהליך המכונה פיסוק. כל פסקה מותאמת למחרוזת מעל א"ב סופי ונבנה מילון בתהליך דינמי.
- כל פעם שמחלקים את המחרוזת מתבצע מעבר על הרשימה המקושרת לראות אם כבר הופיע תת מחרוזת בינארית כזו, אם כן, עוברים על תת מחרוזת גדולה יותר.
- יש להחזיק "מונה" אסקי - שיכיל את הערך האלפבתי הבא הפנוי.

כאשר נגמרים הערכים האלפאבתיים הבודדים נעבור לרצפים מהסוג aa ואז ab ואז ac, כלומר כל הקומבינציות של 2 אותיות, ולאחר מכן כל הקומבינציות של 3 אותיות, וכך הלאה...עד לסיום קריאת המחרוזת הבינארית.

STORIES	TO DO	IN PROGRESS	DONE
היכרות עם סביבת העבודה.	<ul style="list-style-type: none"> הורדה והתקנה של התוכנה. פתיחת פרויקט. 		
קליטת נתונים ושרשור למחרוזת בינארית.	<ul style="list-style-type: none"> גישה לקלט מפרויקט 1. פתיחת הקובץ לקריאה. בדיקת חוקיות של הקלט. יצירת משתנה חדש (מחרוזת) דינאמי שיכיל את המידע שקוראים מהקובץ. קריאת הקובץ עד ל EOF. 		
קוונטיזציה.	<ul style="list-style-type: none"> פתיחת קובץ הפלט שנכתוב אליו את הסדרה האלפביתית. יצירת רשימה מקושרת שתכיל בתוכה תת מחרוזת עם התאמה לערך מעל הא"ב. חלוקת המחרוזת המקודדת הבינארית לתת-מחרוזות הנקראות פסקאות בתהליך המכונה פיסוק. כל פסקה מותאמת למחרוזת מעל א"ב סופי. כל פעם שמחלקים את המחרוזת מתבצע מעבר על הרשימה המקושרת לראות אם כבר הופיע תת מחרוזת בינארית כזו, אם כן, עוברים על תת מחרוזת גדולה יותר. יש להחזיק "מונה" אסקי - שיכיל את הערך האלפבתי הבא הפנוי. שרשור הערכים של הא"ב מהרשימה המקושרת למחרוזת כתיבת המחרוזת של הא"ב לקובץ הפלט 		

כללי



מכיוון שהתנהלות הפרויקט מתוכננת להיות בשיטת סקראם, עליכם ליצור טבלת דרישות **מתועדפת, מוערכת ומשווית**. המקורות לרשימה זו הם: הטבלאות השונות (משימות, דרישות, סיכונים וכדו'), מה-SDP, מה-SRS, מה-SDS, משימות הפרויקט השונות הניתנות בקורס וכל מה שלדעתכם מצריך עבודה של הצוות.



הערכה: כמעט בלתי אפשרי להעריך מאמץ הנדרש למשימה שעוד לא עשינו אף פעם. מצד שני כל מהנדס תוכנה נדרש לתת הערכות. מה עושים? מנסים לחלק את המשימה (לפחות עבור האיטרציה הבאה) עד לרמה שעבורה כבר אפשר לתת הערכה גסה. מתעדים את ההערכות שניתנו ומזינים זמני עבודה כדי שבהמשך נוכל לשפר אותן על סמך הניסיון שנרכש בהערכה.

2. שלבי העבודה

- **מנהל הפרויקט** מכניס אבני דרך לפי תוכנית השחרורים.
- הצוות נפגש ומחליט על משימות בפרויקט (לפי המקורות שלעיל עם דגש על תרחישי שימוש וסיפורי משתמש) ולכל משימה מכניסים אותה לתוכנית עם אחראי (Assign to), סוג (Type), אבן דרך (Milestone), סוג וחשיבות. באזור התוכן של המשימה אפשר להוסיף טקסט נצרך וכמו כן יש לתת תעדוף והערכת זמן בנקודות.
- עבור אבן הדרך הבאה 1 מפרטים משימות (משנה) ומעריכים את המאמץ הנדרש.
- **בתיאום עם הלקוח מחליטים על היקף העבודה לאבן הדרך הראשונה.**

3. בקרת קוד – Source Configuration Management במשימה זו



עליכם להדגים לחברה שהכנסתם כלי כזה לשימוש מלא עבור הפרויקט והוא מכיל כבר שלד של קבצי הפרויקט ולפחות את קבצי נסיון טכנולוגי שביצעתם.

הצעדים הנדרשים:

1. יצירת פרויקט בסביבת העבודה והוספתו למערכת בקרת הקוד.
2. תשתיות שהכנתם מוספים גם כן.
3. אופציה: הוספת ספריה עבור מסמכי הפרויקט (SDF – Software Document Folder) שתנוהל במערכת הגרסאות גם כן.

4. סקר ZFR במשימה זו עליכם להיפגש עם נציגי "הלקוחות" 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 ללקוח שיש לכם שלד שמתקמפל ורץ – אך ייתכן שעדיין אינו מספק שום דרישה פונקציונלית.
- כחלק מהשלד הזה יהיה קובץ Readme (למשל Readme.html) המהווה תיעוד בסיסי לשימוש במוצר שכדרישת מינימום יכיל הערה בדבר רישיון הפרויקט (למשל קוד פתוח BSD עם זכויות מסוימות למכללה).

תיעוד המוצר עליכם ליצור דפי תיעוד למשתמש ולמפתח וכן דף הורדות של גרסה בינארית וגרסת קוד. התיעוד למשתמש יכלול:

- פרטים כללים על המוצר :
 - כיצד להוריד, להתקין ולהריץ את המוצר
 - פרטים ראשוניים על אופן השימוש במוצר.
 - כיצד לדווח על באג שמתגלה (מייל\פתיחת נושא חדש במערכת המשימות וכדו'): פניה למייל alaredo1998@gmail.com ובו יש לפרט את השגיאה
- תיעוד זה יכול להפוך בהמשך למדריך למשתמש, במידה ואתם כבר מכינים מדריך נפרד, תנו קישור אליו.

התיעוד למפתח נועד עבור מפתחים שרוצים להצטרף אליכם או להשתמש בקוד שלכם. הוא יכלול:

- כיצד להשיג את הקוד (גרסה אחרונה וגם ה-repository)
- פרטים על מבנה הספריות והקבצים בפרויקט
- כיצד לקמפל 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)
- כיצד להגיע לרשימת הבאגים וכיצד ניתן לקבל משימה לעבוד על באג.
- קובצי המקור צריכים לכלול הערות, לכל קובץ לפחות שורת הסבר, אם לא טריוויאלי, מה תפקידו.

תוצרי הסקר: עדכון רשימת המשימות בהתאם לתוצאות.

5. תכנון אבן דרך מס' 1

לאחר שהמשימות השונות מוערכות, יש לקבוע פגישה עם נציגי הלקוח בה תבחרו את התרחישים שימושו לאבן דרך 1. משימה שהלקוח מבקש כבר להוסיף היא פיתוח בדיקות יחידה (נלמד תוך כדי העבודה באבן דרך זו).

תוצרים: דף עם כותרת של מטרת האיטרציה הבאה, סיכום פגישת ההכנה, דף משימות אבן דרך זו (sprint log), רישום ברור של סך ההערכה בנקודות למשימות לאבן דרך זו (המספר ישמש אותנו בהמשך). דף זה ישמש כדף של העבודה באיטרציה זו ותוכלו להשתמש בו גם כלוח כללי לשיתוף נושאים בין חברי הצוות.

נספח - צעדים ליצירת פרויקט והכנסתו למערכת בקרת קוד Mercurial בסביבת מיקרוסופט

מקורות: מדריך - <http://mercurial.selenic.com/wiki/Tutorial> - Mercurial
עוד קישורים - <http://mercurial.selenic.com/guide>

1. התקנות: לקוח Mercurial בשם TortoiseHg למעטפת חלונות, אופציה: התקנת לקוח VisualHg והגדרתו בסביבת הפיתוח (קישורים במצגת ובאתר).
2. ביצוע שכפול של המאגר ברשת או פתיחת ספריית עבודה במערכת הקבצים והוספת מאגר (Create Repository Here).
- הערה: גישה לפקודות הלקוח היא ע"י תפריט עכבר ימני במעטפת חלונות ובחירת TortoiseHg, הפעולה המתאימה במדריך זה מופיעה בסוגריים. למעט פעולת היצירה את שאר הפעולות אפשר לבצע גם בתוך סביבת העבודה (למשל Visual Studio) ע"י תפריט עכבר בחלון הקבצים Solution Explorer.
3. הגדרת שם משתמש (Repository Settings -> Edit File) (אפשר גם ישירות בקובץ hg/hgrc או פעם אחת באופן גלובלי בקובץ Mercurial.ini בספריית פרופיל המשתמש).
- [ui]
username = test test@gmail.com
- (הערה: עקב באג לא ניתן לבצע commit אם מבצעים צעד זה פעם ראשונה מתוך חלון ה-commit, במקרה כזה יש לבטל את הפעולה ולבצע שוב).
4. בסביבת הפיתוח: פתיחת פרויקט\ים בספריית העבודה.
(יש לוודא ש-VisualHg מוגדר כספק בקרת הקוד של סביבת העבודה ב: Tools->Options->SourceControl->Current)
5. הוספת קבצי המקור (כולל קבצי סביבת העבודה כגון .sln, .csproj) למאגר (Add + Commit) או בסביבת הפיתוח (Commit) שמזהה את קבצי המקור לאחסון (מומלץ להוסיף גם את הקובץ > .(solution>.sln)
6. סנכרון לשרת בפעם הראשונה, יש להגדיר את השרת
(Configure -> Sync -> Remote Repository -> Add) עם כתובת האזור שלכם בשרת כגון https://robi_y@bitbucket.org/robi_y/se11a, לסיום לדחוף את השינויים (push).
7. (כדאי להתנסות במשיכת קבצי הפרויקט למאגר נוסף המשמש לעבודה שוטפת, כמו בסעיף הקודם או באתר).
8. התנסות בשיתוף פעולה: שליחת בקשה לעדכון\תיקון למשתמש אחר ע"י פתיחת משימה במערכת המשימות (לא קשור לבקרת הקוד)
9. המשתמש בתחנה האחרת שקיבל משימת עדכון: מעבר לספריית היעד ושכפול (clone) בתיבה שנפתחת יש לשים בשדה המקור את כתובת המאגר.
10. פתיחת הפרויקט, ביצוע שינויים (אפשרי ומומלץ בעותק נפרד נוסף), עדכון (commit) וסנכרון לשרת (Synchronize Push) או דרך מייל (Synchronize Email) ישירות למפתח אחר (ודיווח על ביצוע המשימה).
11. המפתח הראשון מבצע pull ו-update כדי לקבל את הגרסה המעודכנת.
12. מומלץ להתנסות גם בחזרה לגרסה ישנה (Update to Revision) וכן במיזוג ופתרון קונפליקטים.

בהצלחה