

## צ'קליסט לסקראם מאסטרים - דוגמא

מייקל ג'יימס

(mj4scrum@gmail.com)

14 לספטמבר 2007

(עודכן ב 2 לפברואר 2016)

### סקראם מאסטר (Facilitator - מסייע ומקל) במשרה מלאה?

הסקראם מאסטר ה"סביר" יכול לשרת שניים או שלושה צוותים בו-זמנית. אם אתה מסתפק בלהגביל את תפקידך לארגון פגישות, הקפדה על מסגרות זמן, ומתן מענה לבעיות/עיכובים ספציפיים שמועלים על ידי חברי הצוות, אזי תוכל להסתדר עם תפקיד הסקראם מאסטר כמשרה חלקית.

הצוות עדיין יראה שיפור יחסית למצב שהיה קיים לפני המעבר לסקראם בארגון, וכנראה שום "אסון" לא יקרה.

אבל אם אתה רואה בעיני רוחך צוות שמצליח לבצע דברים שאף אחד לא חשב שהם אפשריים בארגון העובר שינוי, ועושה את זה בהנאה ובכיף, אתה צריך לשקול להיות סקראם מאסטר מעולה (Great).

סקראם מאסטר מעולה יכול לשרת רק צוות אחד בכל רגע נתון.

אנו ממליצים על הקצאה של סקראם מאסטר אחד לצוות של שבעה, בתור התחלה.

אם עדיין לא גילית מהם כל הדברים שסקראם מאסטר אמור לעשות, פשוט הייה קשוב ל Product Owner) שלך, לצוות שלך, לארגון שמעבר לצוות שלך, והייה מודע לפרקטיקות הטכניות בהן משתמש הצוות שלך.

למרות שאין "מרשם" יחיד שמתאים לכולם – פירמטי להלן נקודות טיפוסיות אותן ראיתי שלסקראם מאסטרים יש נטייה לפספס. אנא סמן כל "קופסה" המיועדת לכך ב-  $\sqrt$ ,  $\Delta$ ,  $?$ , or  $N/A$  כפי שמתואר בעמוד האחרון (דף ההנחיות).

### חלק 1 – איך מתפקד ה Product Owner) שלי?

סקראם מאסטרים יכולים לשפר את יעילות העבודה של ה PO על ידי כך שיעזרו להם למצוא דרכים לתחזק בצורה טובה את הבקלוג של המוצר (Product Backlog), ואת תוכנית שחרור הגרסאות (Release Plan). (שים לב – ה Product Owner הוא האחראי הבלעדי על הבקלוג המתועדף)

☐ האם הבקלוג מתועדף?

☐ האם הדרישות וה"רצונות" של כל בעלי העניין מתועדות בבקלוג? (זכור – הבקלוג הוא "ישות" מתהווה ו"חיה")

□ האם הבקלוג הוא בגודל נוח לתיחזוק/ניהול? כדי לשמר מספר נוח לניהול של פריטים בבקלוג, הקפד שבחלק העליון שלו יהיו פריטים יותר "גרנולריים" (מפורטים וקטנים), ובחלק התחתון – פריטים יותר כלליים וגדולים (Epics). זה לא יהיה "כלכלי" לנתח לעומק פריטים מהחלק התחתון של הבקלוג בשל התיעדוף הנמוך שלהם.

הדרישות ישתנו תוך שיחה מתמדת בין מנהלי המוצר והצוות, לבין בעלי העניין והלקוחות.

□ האם אפשר להגדיר את הדרישות (User Stories) בבקלוג (בייחוד אלה שנמצאות בחלק העליון שלו) כ: בלתי תלויות, שניתן לשאת ולתת לגביהן, בעלות ערך, ניתנות להערכת אומדן, קטנות, וניתנות לבדיקה (INVEST - independent, negotiable, valuable, estimable, small, and) (testable)? (1\*)

□ האם דיברת עם ה PO על "חוב טכני" ועל איך להימנע ממנו? דרך אחת להתייחס לעניין היא להוסיף ל Definition of Done (DOD) של כל פריט בבקלוג: כתיבת בדיקות אוטומטיות ו- ארגון הקוד מחדש (Refactoring).

□ האם הבקלוג זמין ונגיש בצורה קלה לכל בעלי העניין?

□ אם אתה משתמש בכלי אוטומטי לניהול הבקלוג – האם הוא נגיש בקלות וכולם יודעים איך להשתמש בו? הסכנה בשימוש בכלים אוטומטיים היא שהם יכולים עד מהרה להפוך לבלתי נגישים וזמינים לכולם בגלל בעיות הרשאות, ידע בשימוש בכלי וכד'.

□ האם אתה יכול לעזור בהפיכת מידע לזמין על ידי חלוקת תדפיסים לכולם?

□ האם אתה יכול לעזור בהפיכת מידע לזמין על ידי יצירת תרשימים (Charts) גדולים ובולטים?

□ האם עזרת ל PO שלך לארגן פריטים בבקלוג לגירסאות או קבוצות תיעדוף?

□ האם כולם יודעים אם תוכנית הגירסאות עדיין רלוונטית למציאות? אתה יכול לנסות להראות לבעלי העניין תרשים Burndown (2\*) של מוצר/גירסה אחרי שפריטים רלוונטיים סומנו כ "Done" במהלך פגישות ה Sprint Review. הצגה של קצב סיום עבודה של פריטי בקלוג, ופריטים חדשים שנוספו, יכולה לעזור בגילוי מוקדם של סטייה מתכולה/לוח זמנים.

□ האם ה PO שלך עידכן את תוכנית הגירסאות לאחר פגישת ה Sprint Review האחרונה? מעטים הם ה POs (שבדרך כלל משחררים מוצרים בדוקים בזמן) שמעדכנים את תוכנית הגירסה בסוף כל ספרינט (למשל – דוחים תכולה לגירסאות עתידיות בשל תכולה בעדיפות גבוהה יותר שנכנסת).

1\* <http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>

2\* Mike Cohn, *Agile Estimation and Planning*. (2005)

## חלק 2 – איך מתפקד הצוות שלי?

כסקראם מאסטר אתה נדרש "להוביל תוך מתן דוגמא" בעבודה ושיתוף פעולה עם חברי הצוות. יחד עם זאת קיים החשש ש"תלך לאיבוד" במשימות טכניות. לפיכך, קח בחשבון את תחומי האחריות העיקריים שלך לצוות:

☐ האם הצוות שלך במצב של "זרימה" (Flow)? מספר מאפיינים למצב זה (\*3):

- מטרות ברורות (ציפיות ו"כללי המשחק" ידועים והמטרות ברורות-השגה, ומישרות עם יכולות וכישורים של חברי הצוות)
- מיקוד/פוקוס – רמה גבוהה של מיקוד על אזור תשומת-לב מצומצם
- מודעות עצמית פוחתת, ושיפור בעשייה ומודעות צוותית
- פידבק מיידי וישיר (הצלחות וכשלונות תוך כדי עשייה גלויים לעין, כך שההתנהגות יכולה להשתנות על פי הצורך)
- איזון בין האתגר לבין היכולות (פעילויות אינן קשות מדי, וגם לא קלות מדי)
- תחושה של שליטה אישית על המצב או על פעילויות
- הפעילויות מעודדות מוטיבציה פנימית, לכן פחות ניכר מאמץ

☐ האם נראה שחברי הצוות מחבבים אחד את השני, ושמחים בהצלחות אחד של השני?

☐ האם חברי הצוות מצפים אחד מהשני לסטנדרטים גבוהים, ומאתגרים אחד את השני ללמוד ולגדול?

☐ האם יש מקרים בהם הצוות נמנע מדיון בגלל תחושת אי נוחות? (\*4)

☐ האם ניסית לגוון בפורמטים ומקומות בהם נערכות פגישות הרטרו של הצוות? (\*5)

☐ האם הצוות מפוקס על מטרות הספרינט? ייתכן וכדאי לעבור שוב, באמצע הספרינט, על ה Acceptance Criteria של הפריטים בבקלוג עליהם התחייב הצוות לספרינט הנוכחי.

☐ האם לוח המשימות של הספרינט (Sprint Board) משקף נכונה את מה שהצוות עושה? היזהר מ"חומר אפל" של משימות לא ידועות, ומשימות הגדולות מיום עבודה אחד. משימות שאינן קשורות לפריטים בבקלוג עליהם התחייב הצוות מהוות למעשה גורם מעכב לאותן התחייבויות.

\*3 Mihaly Csikszentmihalyi, *Flow: The Psychology of Optimal Experience* (1990)

\*4 Marshall Rosenberg, *Nonviolent Communication: A Language of Life: Life-Changing Tools for Healthy Relationships* (2003).  
Also consider enlisting a professional facilitator who can make uncomfortable conversations more comfortable

\*5 Derby/Larson *Agile Retrospectives: Making Good Teams Great* (2006)

- ☐ האם הצוות מונה 3-9 אנשים, עם מגוון מספק של מיומנויות והתמחויות על מנת שיוכל לבנות מוצר שפוטנציאלית ניתן לדילבור (Potentially shippable product)?
- ☐ האם לוח המשימות (Taskboard) של הצוות מעודכן?
- ☐ האם הפעילויות היומיומיות של הצוות האוטונומי חשופות לכל חברי הצוות, ותוצריהן זמינים לשימוש על ידי כל חברי הצוות?
- ☐ האם התוצרים הנ"ל "מוגנים" מפני גורמים מחוץ לצוות? בחינה מוגזמת של פעילויות יומיומיות של הצוות על ידי גורמים חיצוניים עלולה לפגוע בניהול העצמי והשקיפות הפנימית בצוות.
- ☐ האם חברי הצוות מתנדבים לקחת על עצמם משימות?
- ☐ האם הצורך בטיפול בחוב טכני הועלה ובא לידי ביטוי בצורה ברורה ב Definition of Done על מנת לשפר באופן הדרגתי ונמשך את איכות הקוד?
- ☐ האם חברי הצוות "משאירים את הגדרות התפקיד מחוץ לחדר" ולוקחים אחריות משותפת על מגוון המשימות הנדרשות לביצוע העבודה (בדיקות, תיעוד וכו')?

### חלק 3 – באיזה מצב נמצאות הפרקטיקות הטכניות (Engineering Practices) בצוות?

- ☐ האם למערכת שבפיתוח יש כפתור "העבר לבדיקה" (Push to test) שמאפשר לכולם (בצוות או מצוותים אחרים) לאתר בקלות בעיות רגרסיה שנגרמו על ידם? יכולת כזו מתאפשרת על ידי שימוש בכלים כמו JUnit, NUnit וכד'.
- ☐ האם קיים אצלכם איזון נכון בין בדיקות "מערכת קצה-לקצה" אוטומטיות (end-to-end system tests) ו"בדיקות יחידה" אוטומטיות (unit tests)?
- ☐ האם הצוות כותב את ה- system tests וה- unit tests באותה שפת תכנות בה נכתבת גם המערכת שבפיתוח? שיתוף פעולה בין חברי הצוות לא ישתפר אם ייעשה שימוש בסקריפטים, או בכלים "הקלטה" של פעולות – כלים שרק חלק מחברי הצוות ידעו לתחזק.
- ☐ האם הצוות כבר גילה את התחום האפור השימושי בין system tests ו-unit tests? (\*6)
- ☐ האם שרת continuous integration (\*7) נמצא בשימוש לגילוי אוטומטי והתראה על בעיות רגרסיה? האם ניתן "לכווץ" את מעגל הפידבק הזה לשעות, או אפילו דקות? ("בילדים יומיים" (Daily builds) הם לחלשים" - Kent Beck)

\*6 <http://blogs.collab.net/agile/junit-is-not-just-for-unit-testing-anymore>

\*7 <http://www.martinfowler.com/articles/continuousIntegration.html>

□ האם כל הבדיקות מועלות כחלק מההליך לשרת ה-continuous integration?

□ האם חברי הצוות גילו כבר את חדות ה"עיצוב המתמשך" (continuous design) וארגון הקוד מחדש באופן רציף (constant refactoring) (\*8), כאלטרנטיבה ל"עיצוב גדול מראש" (Big Design Up Front)?

ל"ארגון הקוד מחדש" (refactoring) יש הגדרה ברורה: שינוי מבנה פנימי בלי לשנות התנהגות חיצונית. refactoring צריך לקרות מספר פעמים בשעה, בכל מצב בו יש שכפול של קוד, לוגיקה מורכבת מותנית (העמדה מוגזמת או מתודות ארוכות מדי), מזהים עם שמות לא משמעותיים, צימוד מוגזם בין אובייקטים, וכד'. ביצוע של refactoring בצורה "בטוחה" יכול לקרות רק עם כיסוי של בדיקות אוטומטיות. המנעות מביצוע refactoring יקשה יותר ויותר על ביצוע שינויים במוצר בעתיד, בייחוד בגלל שנהיה קשה למצוא מפתחים טובים שיהיו מוכנים לעבוד על קוד גרוע.

□ האם ה-Definition of Done לכל פריט בבקלוג כולל כיסוי מלא של בדיקות אוטומטיות ו-refactoring? לימוד הפרקטיקה של Test Driven Development (TDD) יכול להעלות את הסבירות להשיג זאת.

□ האם חברי הצוות משתמשים רוב הזמן ב"עבודה בזוגות" (pair programming)? עבודה בזוגות יכולה לשפר בצורה משמעותית את התחזוקתיות של הקוד ולהוריד את כמות הבאגים. היא מאתגרת גבולות ולעיתים נראית כמו בזבז זמן (אם מודדים שורות קוד שנכתבות ולא פונקציונליות שניתן להעביר ללקוח). מומלץ ליזום "ימי עבודה בזוגות" בצוות. יש סיכוי גבוה שלפחות חלק מחברי הצוות יתחילו להעדיף לעבוד בצורה הזו.

#### חלק 4 – איך מתפקד הארגון?

□ האם יש תקשורת בין-צוותית במידה מספקת? פגישות "Scrum of Scrums" הן דרך אחת להשיג זאת (\*9)

□ האם צוותים יכולים לייצר פיצ'רים עובדים בצורה עצמאית, אפילו אם אלה חוצים גבולות ארכיטקטוניים? (\*10)

□ האם הסקראם מאסטרים בארגון נפגשים אחד עם השני, לעבוד על רשימת הליקויים (impediments) הארגונית?

\*8 Martin Fowler, *Refactoring: Improving the Design of Existing Code* (1999)

\*9 See <http://less.works/less/framework/coordination-and-integration.html> for alternatives

\*10 <http://FeatureTeamPrimer.org>

□ כשמתאים ומקובל, האם רשימת הליקויים הארגונית תלויה על קירות משרדי מנהלי הפיתוח הבכירים? האם ניתן לכמת את עלותם של הליקויים בדולרים, time to market אבוד, איכות נמוכה, או אובדן הזדמנויות לקוח? (אבל זכור תמיד ללמוד מהטעויות של Ken Schwaber: "A dead ScrumMaster is a useless ScrumMaster" (\*11))

□ האם הארגון שלך הוא מהמעטים עם מסלול קריירה שמתאים למטרות הקולקטיביות של הצוותים בארגון? ענה ב- "לא" אם יש תמריץ קריירה (\*12) לעבודת תכנות או ארכיטקטורה על חשבון בדיקות, אוטומציה של בדיקות, או תיעוד למשתמשים.

□ האם הארגון שלך הוכר על ידי העיתונות המסחרית או מקור עצמאי אחר כאחד הארגונים שהכי טוב לעבוד בהם, או כמוביל בתעשייה שלו?

□ האם אתם מייצרים "ארגון לומד" (A learning organization)?

### מסקנה

אם אינך יכול לסמן את רוב הנקודות הנ"ל כ-√ ("עובד היטב") ועדיין יש לך מעט זמן חופשי במהלך היום – הייתי שמח לשמוע ממך.

לא קיימת נוסחה ברורה ליצירת כושר המצאה ותבונה אנושית. מסמך זה מעלה נקודות שיכולות לעזור, או שלא, בסיטואציה שלך.

אבל ברגע שהתחלת להבין מה אתה יכול לעשות כדי לגרום לשינוי, אתה עלול למצוא את עצמך מפחד לעשות אותו. זה הסימן שאתה בכיוון הנכון.

\*11 Ken Schwaber, *Agile Project Management with Scrum* (2004)

\*12 Alfie Kohn, *Punished By Rewards: The Trouble with Gold Stars, Incentive Plans, A's, Praise, and Other Bribes* (1999)

## **טופס ליקויים ארגוני - Organizational Impediment Form**

הבעיה הנראית לעין (Surface issue):

שורש הבעיה (השתמש ב- "למה?" חמש פעמים) (Root cause (Use five times "Why?")):

אימפקט עסקי (Business Impact):

אימפקט ריגשי (Emotional Impact):

פעולה נדרשת (Clear, actionable request):

**(העתק דף זה כמספר הפעמים הנדרש)**

## דף הנחיות

אם קיבלת את הצ'קליסט הזה כמשימת תרגול, והמעסיק הנוכחי שלך (או זה שלפניו) ניסה ליישם משהו שנראה כמו סקראם, אנא ענה על הנקודות בצ'קליסט על פי מה שראית שם.

סמן כל אייטם בצ'קליסט עם אחד מהבאים:

✓ - "עובד היטב"

Δ - "יש מקום לשיפור ואני יודע איך להתחיל"

? - "יש מקום לשיפור, אבל איך?"

N/A - "לא רלוונטי" או "לא יביא שום תועלת"

או, אם המעסיק הנוכחי שלך (או זה שלפניו) לא ניסה ליישם משהו שנראה כמו סקראם, סמן כל אייטם בצ'קליסט עם אחד מהבאים:

✓ - "עובד היטב" או "יהיה קל לגרום לזה לעבוד היטב"

Δ - "יהיה מאתגר ואני יודע איך להתחיל"

? - "יהיה מאתגר ואין לי מושג איך להתחיל"

N/A - "לא רלוונטי" או "לא יביא שום תועלת"

אחרי שסימנת את כל האייטמים בצ'קליסט, בחר 2-6 ליקויים ארגוניים (Impediments) ומלא בהתאם את טופס הליקויים הארגוני הנ"ל (העתק אותו כמספר הליקויים שבחרת), אם הם נגזרים מהצ'קליסט ואם לאו.  
בחר בליקויים שיש לך סיכוי (או תקווה) של לפחות 1% לשנות.