



ייזום, תכנון  
והתארגנות

הקדמה

התקנה,  
ייצור ותפעול

פיתוח  
ובדיקות

# הקדמה

- ◀ המטרה העיקרית של כל פרויקט הינו פיתוח תוצרים רלוונטיים ומתאימים ללקוח בזמן הקצר ככל הניתן, באיכות הנדרשת ובתקציב הקיים.
- ◀ כתוצאה מהצורך לספק פתרונות מהירים ולהיענות לשינויים ודרישות חדשות לבקרים, פותחו שיטות AGILE שונות המכילות אוסף של עקרונות כלליים (ראה עקרונות כלליים **MethodAgile**)



גמישות לשינויים  
על פני היצמדות  
לתכנית



שיתוף הלקוח  
על פני מו"מ חוזי



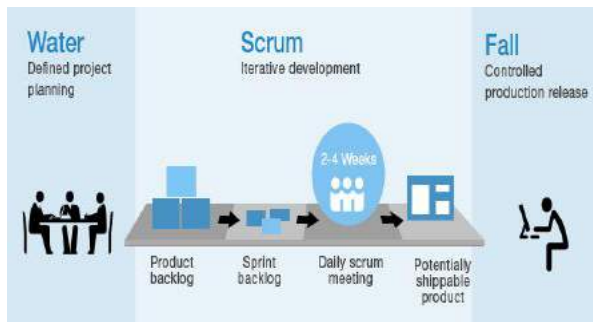
תוכנה עובדת  
על פני תיעוד נרחב



אנשים ויחסי גומלין  
על פני תהליכים וכלים

# יישום האגיל

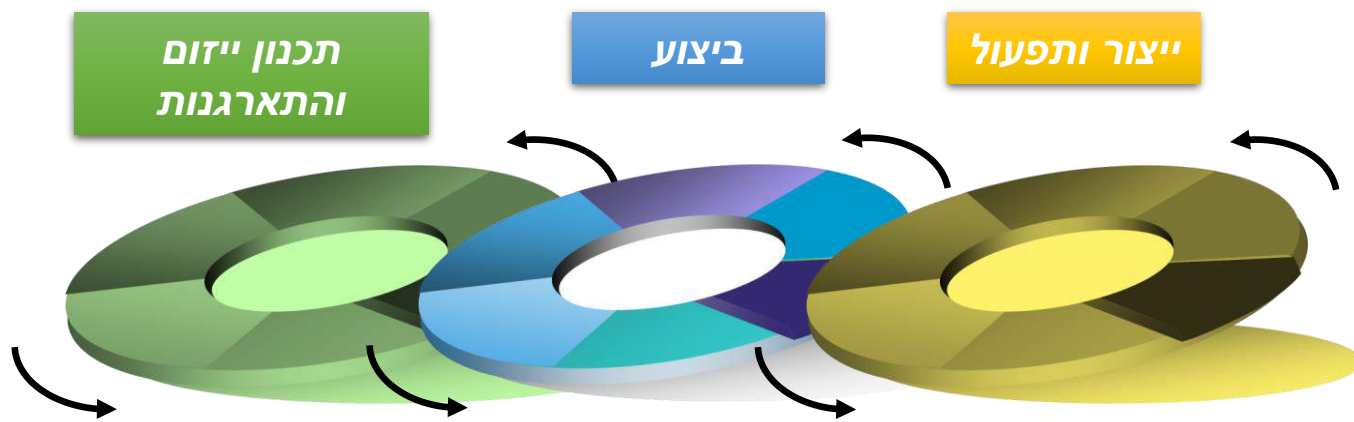
יישום תהליכי אגיל נמצא על ספקטרום החל מיישום חלקי ועד יישום מלא. כל ארגון / חברה חייבת להתאים את מודל האגיל כתלות בסוג הפרויקט, הלקוחות וכוח האדם, שיטות ניהול הפרוייקט ומאפיינים נוספים.



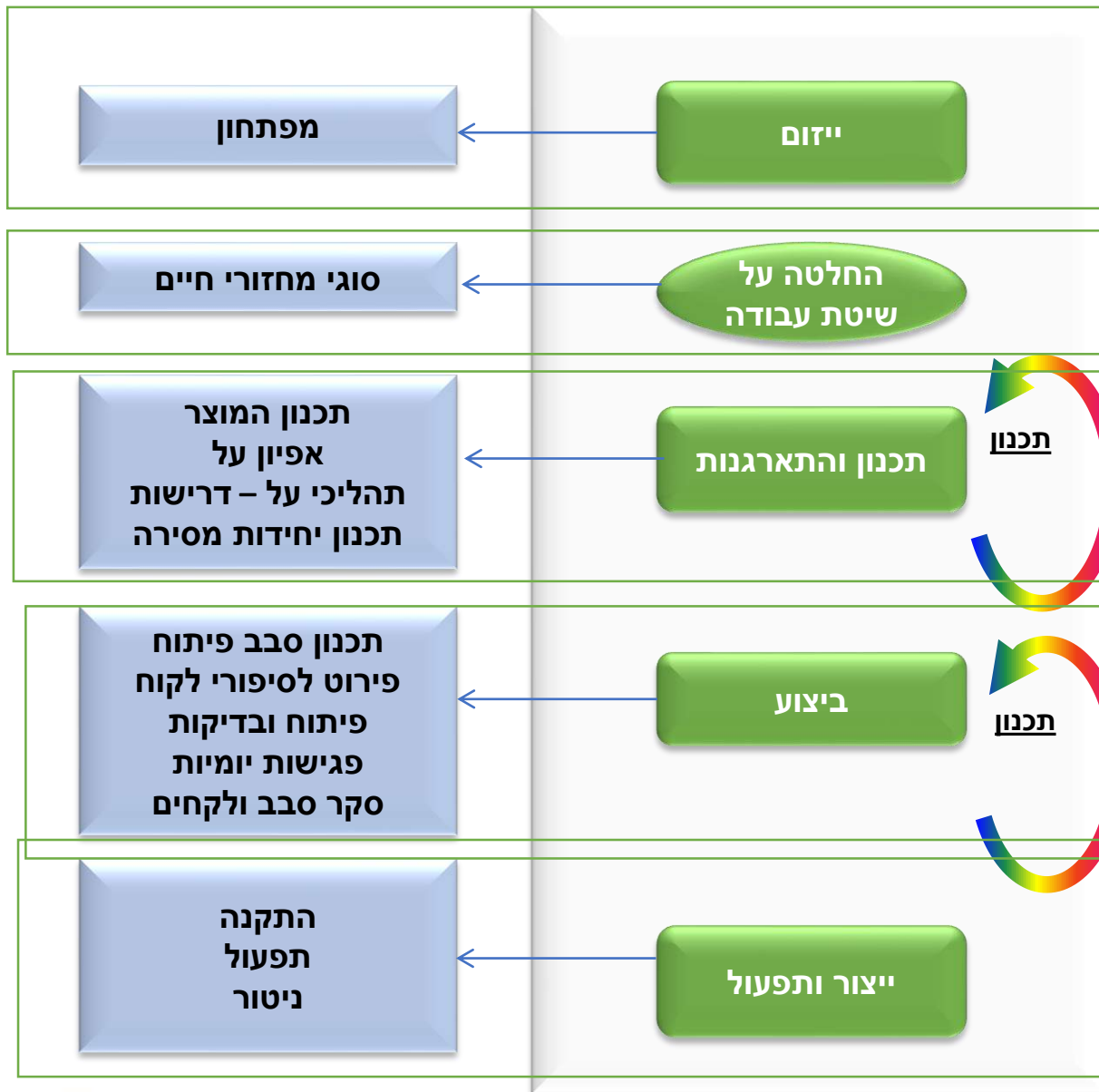
# מחזור חיים אג'ילי

◀ מחזור החיים של מוצר מורכב ממספר "פעילויות אב". כאשר אופן היישום שלהם תלוייה ברמת האג'יליות בפרויקט.

## מחזור החיים מורכב מסדרה של פעילויות מתמשכות *Continuous All*



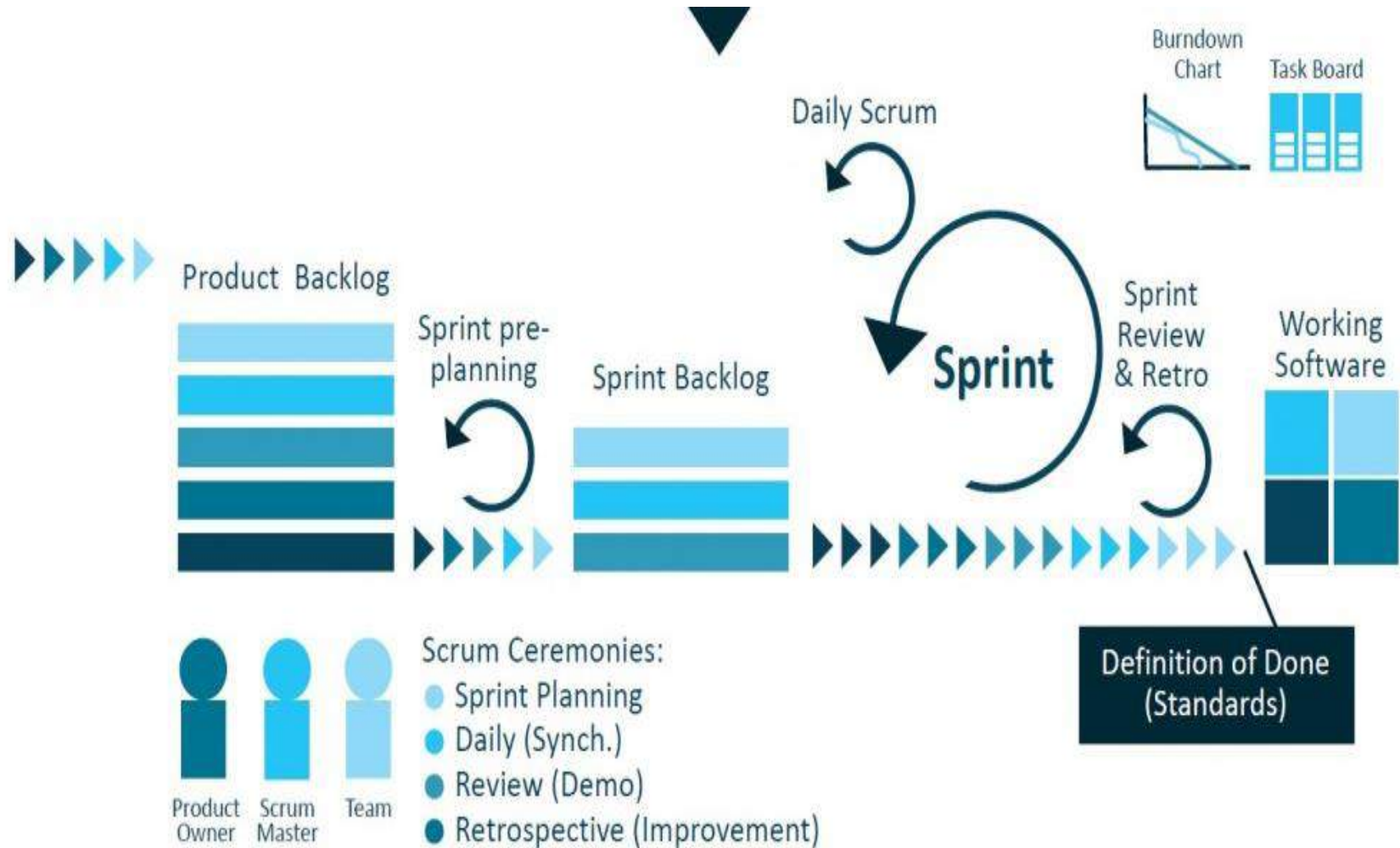
# מבט על



תיעוד  
 שערי פרויקט  
 סקרים  
 בקרה  
 סיכונים  
 תבניות  
 בדיקות  
 ניתוח דרישות  
 תכנית עבודה  
 מכרזים  
 ספק לקוח

כלים לניהול

# Scrum in a Nutshell



ייזום, תכנון  
והתארגנות

הקדמה

התקנה,  
ייצור ותפעול

פיתוח  
ובדיקות



# ייזום תכנון והתארגנות

## Planning – תכנון

▶ גיבוש דרישות עסקיות, בדיקת היתכנות טכנולוגית וניתוח חלופות (במידת הצורך).

▶ קביעת שיטת הפיתוח, אופן הפיתוח, צוות הפיתוח, כלים ומשאבים, סטנדרטים, טכנולוגיה, משאבים, מסגרת לו"ז, הערכת סיכונים, תהליך הבקרה וההערכה.

מפת"חון



▶ ראה: סוגי מחזורי חיים אגיליים

# ייזום תכנון והתארגנות

## אפיון על

- ◀ רשימת תכונות והפונקציות הנדרשות מהמוצר, כולל עדיפות, מורכבות פיתוח והערכת מאמץ נדרש (בדרך ב-story points או בימי עבודה).
- ◀ **Product Backlog ראשוני**. אוסף הדרישות ב- *Product Backlog* נבדק ומעודכן לאורך כל הפרויקט.
- ◀ *Architecture/High level Design* - הגדרה ראשונית של ארכיטקטורת המוצר. הגדרה זו תעודכן עם ההתקדמות בפיתוח, הבנה מעמיקה יותר של הדרישות, וקבלת דרישות חדשות שלא היו ידועות מראש.

אפיון אג'יל הינו תהליך מתמשך ואבולוציוני  
*Continuous Analysis*

# ייזום תכנון והתארגנות

## תוכן העניינים של אפיון-על מינימלי

### 1 יעדים

1.1 מבוא

1.2 יעדים כללים

1.3 מטרות מדידות

1.4 לקוח, מומחה היישום וקהל יעד

1.5 בעיות שהמערכת תפתור

### 2 יישום

2.1 מילון מונחים

2.2 תרשים ישויות

2.3 Epic (הסבר על נושא ה-epics

(בהמשך)

2.4 Epic – תת מערכת/ נושא עיקרי

...

2.9 ממשקים (אופציונאלי)

2.10 אבטחת מידע

3 טכנולוגיה

4 מימוש

4.1 גורמים מעורבים

4.2 שלבי מימוש

5 עלות

### אפיון על מבוסס תבנית

◀ המאפיין יוצר מסמך אפיון-על חדש

בהתבסס על תבנית ארגונית.

◀ קיימות במתודה דוגמאות רבות של

תבניות מורחבות יותר למערכות

מורכבות/גדולות או לנושאים טכנולוגיים

ספציפיים

◀ ניתן לייצר תבנית מותאמת לכל סוג

פיתוח.

◀ יצירת אפיון על יכול להתבצע בעזרת

כלי כגון Confluence.

### דוגמה לאפיון-על

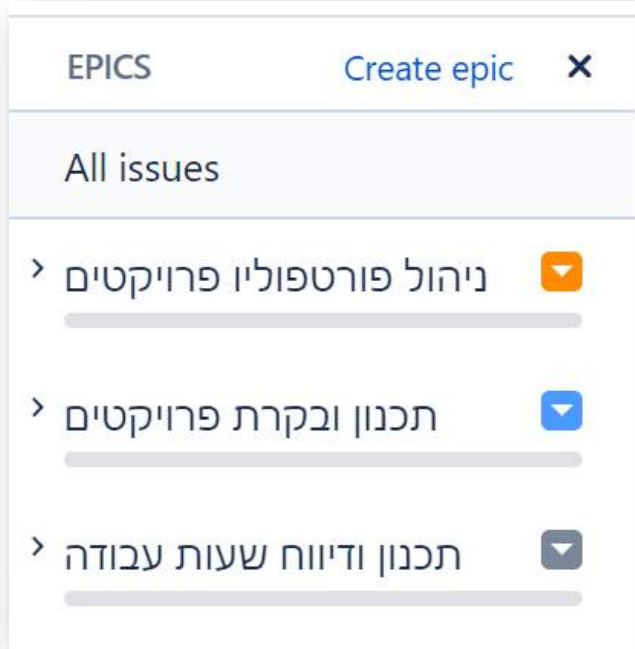


Microsoft Word

Document

# ייזום תכנון והתארגנות

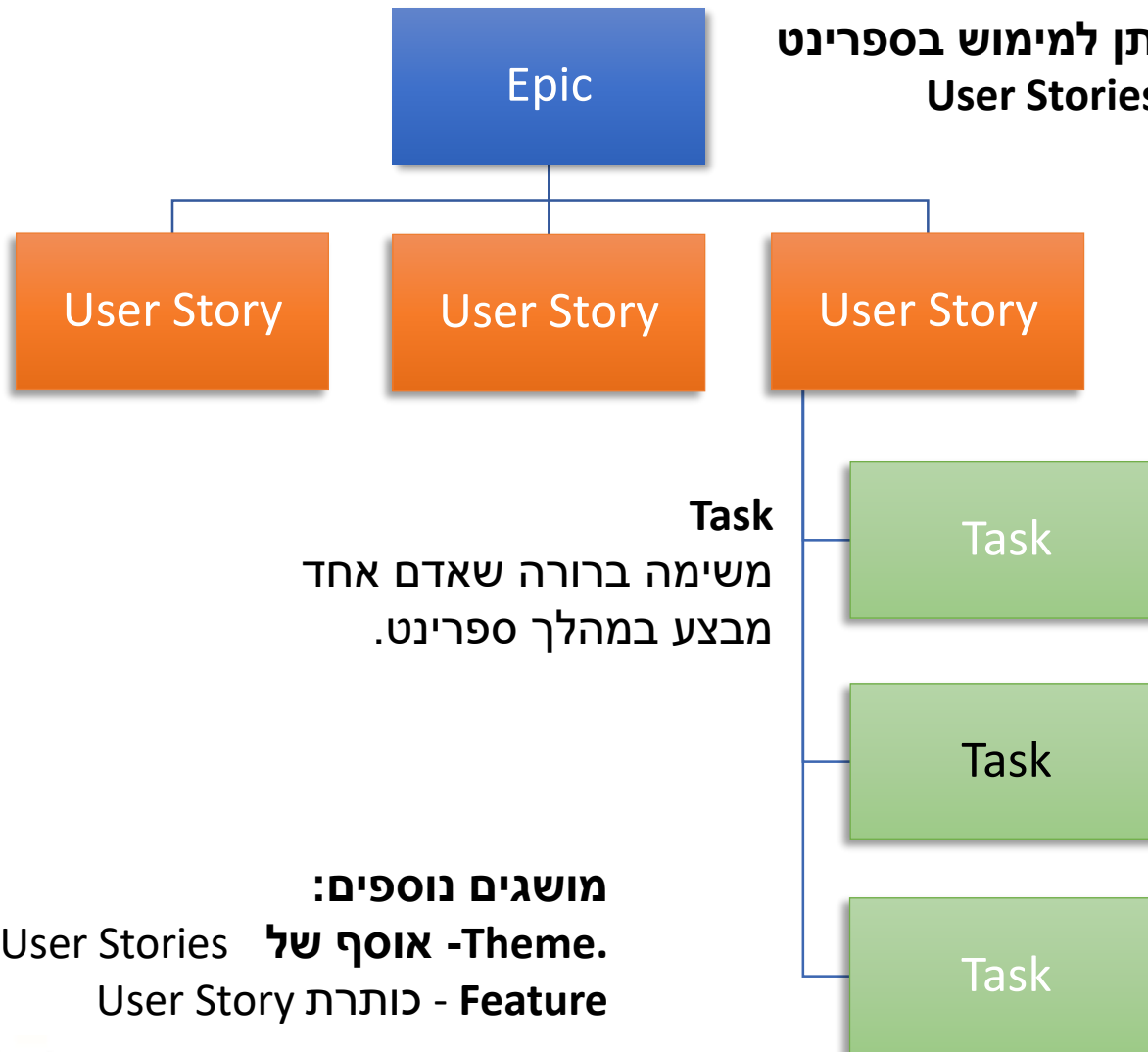
## Epics - תת מערכת



- ▶ Epic הוא נושא רחב הקיים באפיון ומאגד תחתיו אוסף של User Stories.
- ▶ אפיון-העל מכיל רשימה ברורה של Epics.
- ▶ כל נושא גדול באפיון-העל יהיה Epic. למשל: במערכת ניהול רכש, ניתן ליצור Epic בשם "הזמנות"
- ▶ בדרך כלל מומלץ ליצור לכל ישות בתרשים הישויות Epic משלה (אך לא תמיד, תלוי בהקשר).
- ▶ ניתן גם ליצור Epics שאינם פונקציונאליים, כגון: ביצועים, אמינות, יציבות, וכו'.
- ▶ תבנית אפיון-העל מכילה את רשימת ה-Epics הלא פונקציונאליים שאמורים להופיע בדרך כלל. המאפיין יוסיף/יעדכן את הרשימה לפי הצורך.
- ▶ לכל Epic מומלץ ליצור דף ב-Confluence שמתאר אותו.

## פירוק ה- Epics ל- User Stories

סיפור-על. לא ניתן למימוש בספרינט אחד. אוסף של User Stories



### User Story

פיסת פונקציונאליות ברורה ובעלת-ערך, הניתנת למימוש במסגרת ספרינט אחד.

**DoD** - Definition of Done - User Story מגדיר בבירור מתי נחשב ככזה שסופק בהצלחה.

### Task

משימה ברורה שאדם אחד מבצע במהלך ספרינט.

מושגים נוספים:

Theme - אוסף של User Stories  
Feature - כותרת User Story

# ייזום תכנון והתארגנות

## אפיון-העל עם רשימת Epics ו- Stories

◀ כך רואה המאפיין את אפיון-העל:

2.5 תכנון ודיווח שעות עבודה

TO DO SHEFA-12 - תכנון ודיווח שעות עבודה

Description	Labels	Summary	Status	Sprint	Key	T
עדכון ימי עבודה בחודש עבור כלל העובדים (ימי תקן) תוך סימון ימי חג וערבי חג		<a href="#">הגדרת לוח שנה</a>	TO DO	SHEFA Sprint 1	SHEFA-31	■
דוח שיראה לזמן הקרוב - לפחות שבוע קדימה - כמה שעות פנויות יש לעובד שלא תוכננו עדיין - לפי התכנון המעודכן האחרון	דו"חות	דוח זמינות עובד לתקופה הקרובה - לפחות שבוע קדימה	TO DO		SHEFA-32	■
עובד יכול לדווח שעות ולרשום שהשעות ידווחו ע"י עובד אחר. בסוף החודש ניתן יהיה להוציא דוח שעות כולל ללקוח, הכולל את השעות בחלוקה לעובדים, כאשר השעות שנרשמו על שם עובד אחר יהיו רשומות על שם העובד האחר ולא על שם העובד שדיווח אותן.		דיווח שעות בשם עובד אחר	TO DO	SHEFA Sprint 2	SHEFA-36	■

◀ זהו אפיון-על **חי**. זהו מסמך מתגלגל שמראה בכל רגע נתון את המצב האמיתי של הפיתוח.

◀ מרגע שה- Epics וה- Stories הוגדרו בכלי ממוכן כל ההתנהלות והשינויים מבוצעים בכלי וניתנים לצפייה דרך מסמך האפיון, תוך הבנת ההקשר הכללי ליעדי המערכת ולתרשים הישויות.

## **Release Planning – תכנון מהדורות**

- ▶ **תכנון ואופטימיזציה של תכולות לגרסה - הקצאת כל Product Backlog Item למהדורה מתוכננת של המוצר, כדי לייצר roadmap ראשוני הכולל תכולות, לוח זמנים לשחרור גרסאות (לפחות הגרסאות הראשונות), ומשאבי פיתוח נדרשים.**
- ▶ **מנהל הפרויקט/המוצר מגדיר את גרסאות המוצר השונות, את תאריך השחרור הרצוי לכל גרסה ומקצה Stories לגרסאות.**
- ▶ **בעזרת **JIRA Portfolio** ניתן לתכנן תכנית אופטימלית הלוקחת בחשבון תכולות (Stories) מסודרות לפי עדיפות, זמינות אנשים מתאימים, תלויות לוגיות בין תכולות ותאריכים רצויים לשחרור גרסאות.**

הקדמה

ייזום, תכנון  
והתארגנות

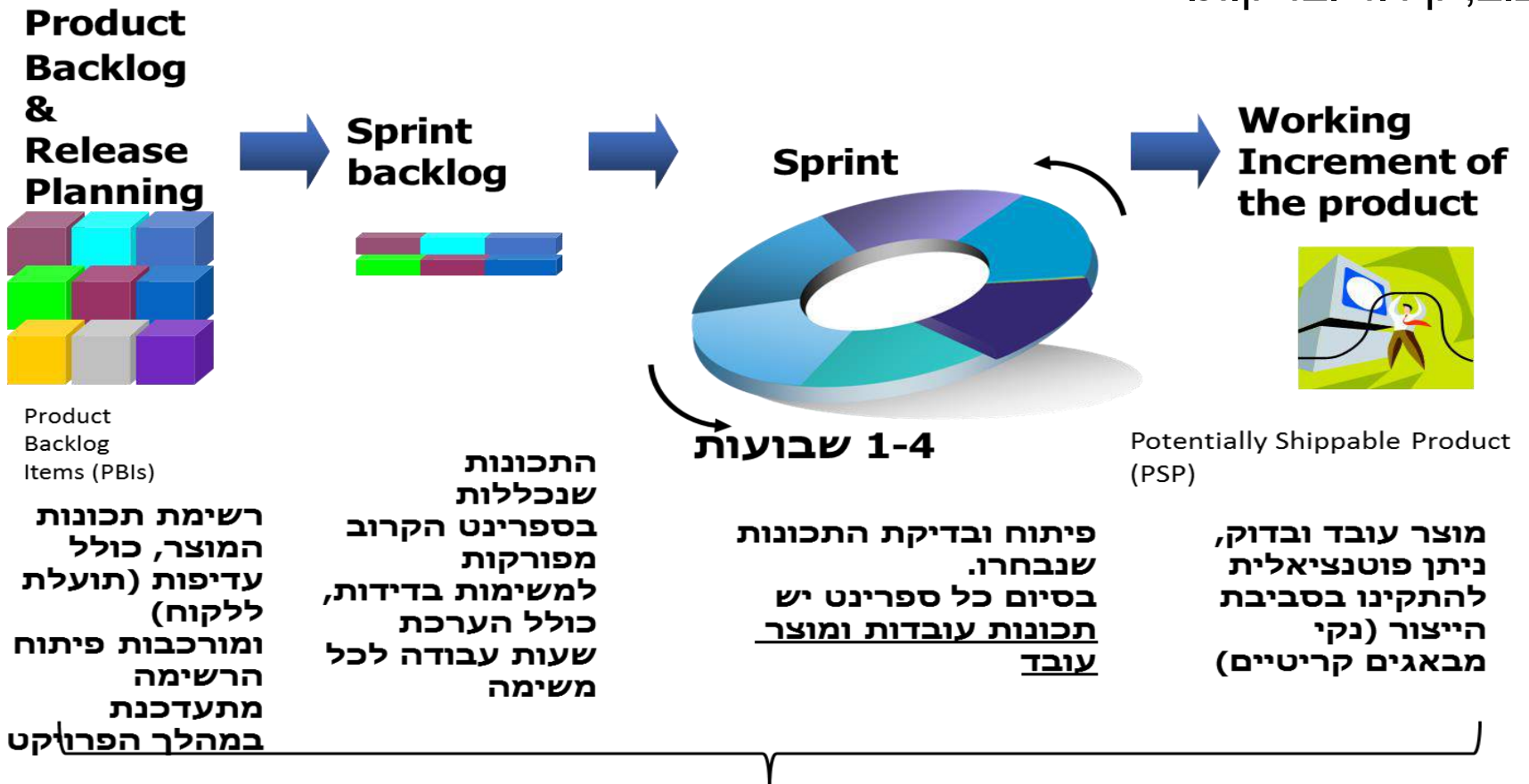
התקנה,  
ייצור ותפעול

פיתוח ובדיקות



# פיתוח בספרינטיים - Development-

Sprint הינו מאמץ מוגבל בזמן בד"כ נמשך כ 2-4 שבועות ומכיל פיתוח התכונות שנבחרו ב . Sprint Backlog במהלך הספרינט המוצר עובר את כל שלבי הפיתוח: אפיון מפורט, עיצוב, קידוד ובדיקות.



# פיתוח בספרינטיים - Development

## Sprint Backlog Planning

- ◀ החלטה על רשימת ה- User stories / Backlog Items הנכללים בספרינט הקרוב, כולל דרישות פונקציונליות, דרישות טכניות ותיקוני שגיאות.
- ◀ הדרישות שנכללות בספרינט הקרוב מפורקות ל- Tasks. משימות בדידות, כולל (אופציונלית) הערכת מאמץ בשעות/ימי עבודה לכל משימה.
- ◀ הגדרת המשימות
- ◀ הערכת כמות שעות נדרשת
- ◀ זיהוי תלויות וסיכונים
- ◀ הקצאת משימות הפיתוח לאנשי הצוות
- ◀ רישום מסודר ב-Sprint Backlog

# פירוק ה- Epics ל- User Stories

- ◀ מנהל הפיתוח מקצה כל Story לאחד מצוותי הפיתוח.
- ◀ כל Story שהוקצה לטיפול צוות פיתוח יופיע ב-Backlog של אותו צוות, וצוות הפיתוח ימלא לו הערכת מאמצים (Estimation).

The screenshot displays the Jira interface. On the left, a backlog view shows a list of issues. The top issue is 'תכנון לויז פעילויות בפרויקט' (SHEFA-27) with a 'תכנון ובקרת פרויקטים' label. Below it is 'דשבורד לפרויקט' (SHEFA-34) with a 'קרת פרויקטים' label. The bottom issue is 'הערכת שעות עבודה לפעילויות בפרויקט' (SHEFA-26) with a 'תכנון ובקרת פרויקטים' label. The right panel shows the details for issue SHEFA-26, titled 'הערכת שעות עבודה לפעילויות בפרויקט'. A red box highlights the 'Estimate' field, which is currently empty. Below it, the 'Remaining' field shows 'Unestimated'. The 'Attachments' section is empty, and the 'Sub-Tasks' section shows 'There are no sub-tasks'.

◀ עקרונית מומלץ לנסח User Story לפי התבנית הבאה: כ- <סוג משתמש> אני רוצה <יכולת> על-מנת <סיבה> למשל: ▶

◀ Epic: כסוכן מכירות, אני רוצה שהמערכת תאפשר לי לבטל פעולות גדולות שהתחלתי, כל עוד הן עדיין מתבצעות, כדי שאוכל להתחרט ולעשות פעולות אחרות בלי לחכות לסיום פעולות שהתחלתי כבר.

◀ User Story: כסוכן מכירות, אני רוצה שהמערכת תאפשר לי לבטל הפעלה של דו"ח הזמנות חודשי כל עוד הוא מתבצע, כדי שאוכל להתחרט ולעשות דברים אחרים (אם לוקח יותר מדי זמן להפיק אותו).

## חיבוריות בין אפיון-העל והפיתוח

- ◀ קיים קשר הדוק בין אפיון-העל לפיתוח והסיכון ל-"התבדרות" ואי-עמידה ביעדי הפרויקט קטן בעת שימוש בכלי ממחושב לניהול.
- ◀ המאפיין רואה את אפיון-העל עם כל ה-Epics וה-Stories בדיוק כפי שצוות הפיתוח רואה אותם.
- ◀ הצוות רואה את זה בתוך ה-Scrum Board שלו (נושא ה-Scrum Board מוסבר בהמשך).
- ◀ בצורה זו, הסיכוי ל-"נתק" בין האפיון והפיתוח קטן.
- ◀ [להרחבה, ראה תהליך מלא.](#)

# המחשת חיבוריות ועקיבות

## User Stories, Tasks, מערכת,

T	Key +	Summary	Labels	Assignee	P	Status	Updated	Due	Epic Link	Epic Name
	MEMO-1	Story A	UI	Unassigned		TO DO	20/Mar/15		Manage different To-Do lists	
	MEMO-2	Story B	UI	Unassigned		TO DO	20/Mar/15		Manage different To-Do lists	
	MEMO-3	Story C	Functionality	Unassigned		TO DO	20/Mar/15		Manage different To-Do lists	
	MEMO-4	Story D	Functionality	Unassigned		DONE	10/Jan/15		Manage reminders of tasks	
	MEMO-5	The ABC App shall support swipe down gesture to delete a picture.	Functionality Gestures	Unassigned		IN PROGRESS	04/Jan/15			
	MEMO-6	RISK 1 - Mitigation 1	None	Unassigned		TO DO	04/Jan/15			
	MEMO-7	RISK 1 - contingency 1	None	Unassigned		TO DO	10/Jan/15			
	MEMO-8	Manage reminders of tasks	Functionality	Unassigned		TO DO	20/Mar/15			Manage reminders of tasks
	MEMO-9	Manage different To-Do lists	Functionality	Unassigned		TO DO	20/Mar/15			Manage different

עקיבות  
ברורה



## התיעוד

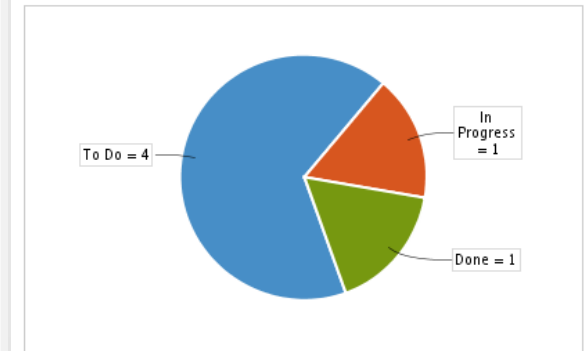
### Application

#### Functionality - Epics

The main functionality of the app is:

MEMO-8 - Manage reminders of tasks TO DO

MEMO-9 - Manage different To-Do lists TO DO



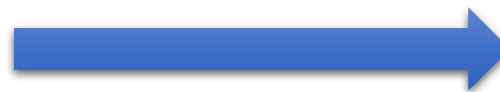
משתקפת ברשימת הדרישות בכלי

הוספת דרישה חדשה בתיעוד



חדש או User Story הוספת  
דרישה עסקית חדשה

משתקפים אוטומטית בתיעוד



## ביצוע הספרינט

- ▶ פיתוח תכולת הספרינט, אינטגרציה ושילוב רכיבים שפותחו עי צוותים אחרים.
- ▶ בדיקת מערכת, עומסים ורגרסיה (רצוי כמה שיותר בצורה אוטומטית), בדיקות בסביבת Pre-production/Staging (בעיקר בספרינטים האחרונים).
- ▶ פיתוח ובדיקת הסבות נתונים (אם רלוונטי).
- ▶ פיתוח ובדיקות ואריזת המערכת, תהליך ההתקנה.
- ▶ עדכון תיעוד נלווה (בעיקר לקראת הספרינטים האחרונים)



# עקרונות אג'יל לבדיקות תוכנה

## The TESTING Manifesto

we value:

- Testing throughout **over** at the end
- Preventing bugs **over** finding bugs
- Testing understanding **over** checking functionality
- Building the best system **over** breaking the system
- Team responsibility for quality **over** tester responsibility





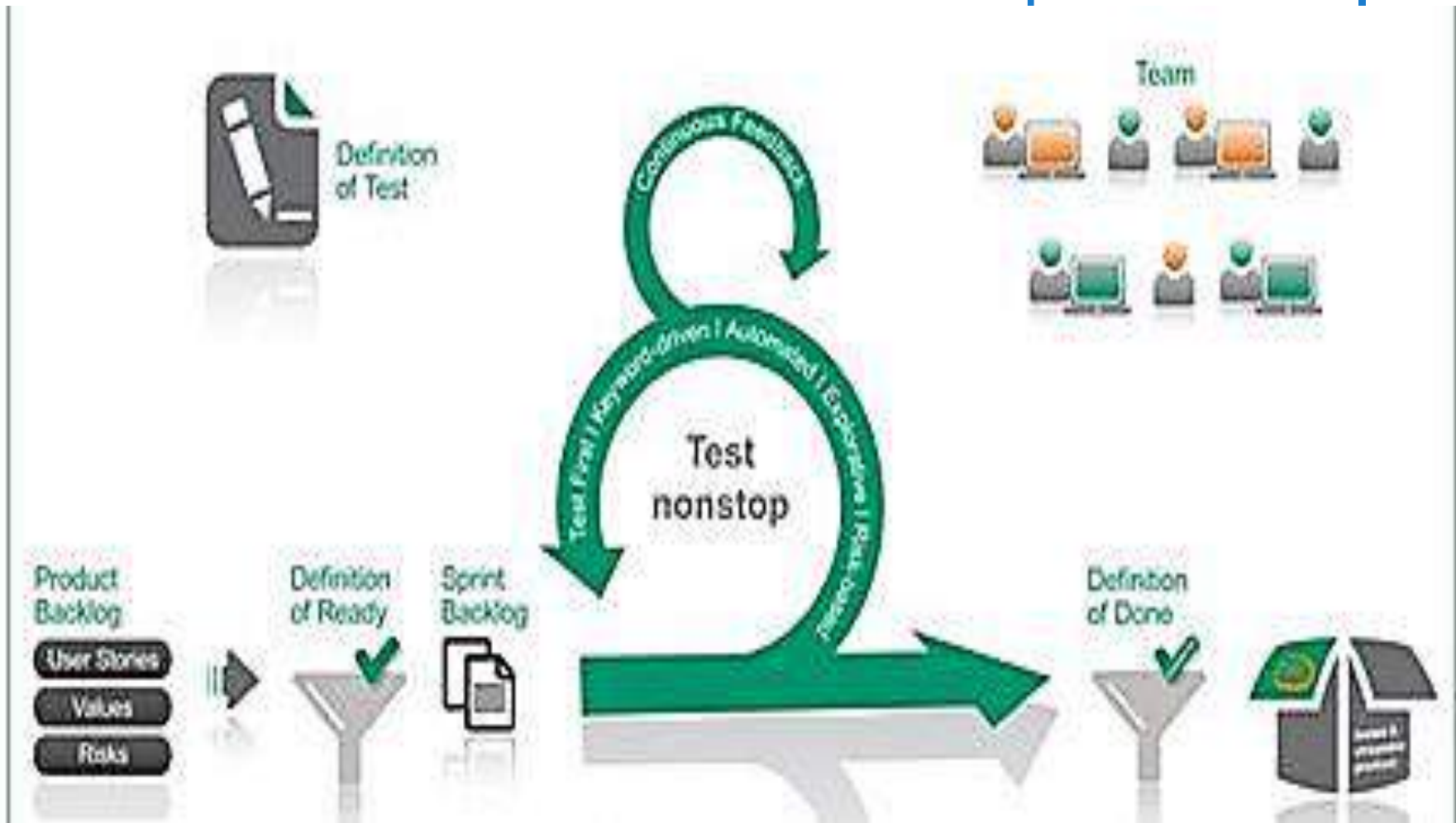
# תכנון וביצוע בדיקות

צוות הבדיקות מתכנן ומבצע בדיקות אוטומטיות (ידיניות במידת הצורך) על המוצר תוך שימוש בכלי ניהול בדיקות.

The screenshot displays the JIRA interface for test management. On the left, a 'Summary' sidebar lists test execution steps for 'Test Plan XRAY-23' and 'Test Plan XRAY-24', including options like 'Add attachment test', 'full body scan', and 'Stop in a middle of scan action'. The main area shows 'Tests' with an 'Overall Execution Status' bar and a summary: '1 PASS 2 FAIL 3 ABORTED' and 'TOTAL TESTS: 6'. Below this, a table lists test sets with their statuses: XRAY-19 and XRAY-20 (PASS), XRAY-19, XRAY-20, and XRAY-21 (ABORTED), and XRAY-19 and XRAY-20 (ABORTED). The bottom left shows 'Test Details' for a specific test step, including test data and results.

#Def	Test Sets	Assignee	Status
2	XRAY-19 XRAY-20	Mor Turgeman	PASS
0	XRAY-19 XRAY-20 XRAY-21	Mor Turgeman	ABORTED
0	XRAY-19 XRAY-20	Mor Turgeman	ABORTED

# תכנון וביצוע בדיקות



The screenshot displays the JIRA Test Management interface. At the top, there's a navigation bar with 'JIRA', 'Dashboards', 'Projects', 'Issues', 'Tests', and 'Create'. Below this, the main header shows 'Star Battles / SB-3' and the issue title 'Verify the app can be installed from the AppStore'. A 'Store' button is visible. The left sidebar contains 'Details', 'Description', and 'Test Details' sections. The main content area is titled 'Test Cycles' and shows a 'Cycle Summary' for 'Functionality' tests. It includes a table of test results with columns for ID, Status, Summary, Detect, Component, Label, Executed By, and Executed On. Below the table, there are summaries for 'Regression' and 'Final Sanity Tests' cycles.

**Functionality Cycle Summary:**  
 Started On: 18/Mar/12  
 Build: 766  
 Environment: QA1  
 Progress: 68.18%

ID	Status	Summary	Detect	Component	Label	Executed By	Executed On
IC-1	FAIL	Verify the app can be installed from the AppStore	-	Main screen	installation	Zephyr Admin	04.05.12 18:04
IC-3	PASS	Verify the app can be launched	-	Main screen	installation	Zephyr Admin	04.05.12 18:05
IC-6	PASS	Verify app starts cleanly	-	Main screen	installation	Zephyr Admin	10.05.12 15:53
IC-7	BLOCKED	Verify app can be closed	-	Main screen	installation	Mike Belts	08.12.11 12:24
IC-8	FAIL	Verify all the buttons on the app show up	-	Main screen	installation	Lana Malakova	08.12.11 12:13
IC-9	PASS	Verify all the images show cleanly	-	Main screen	installation	Mike Belts	08.12.11 12:07
IC-10	UNEXECUTED	Verify that the app can be upgraded	-	Main screen	upgrades		
IC-12	PASS	Verify app can be dimmed	-	Main screen	screen_controls	Lana Malakova	08.12.11 15:01
IC-13	PASS	Verify app can put in hibernation	-	Main screen	screen_controls	Mike Belts	08.12.11 15:41
IC-14	PASS	Verify that the app can have different backgrounds	IC-11			Zephyr Admin	21.01.12 19:56

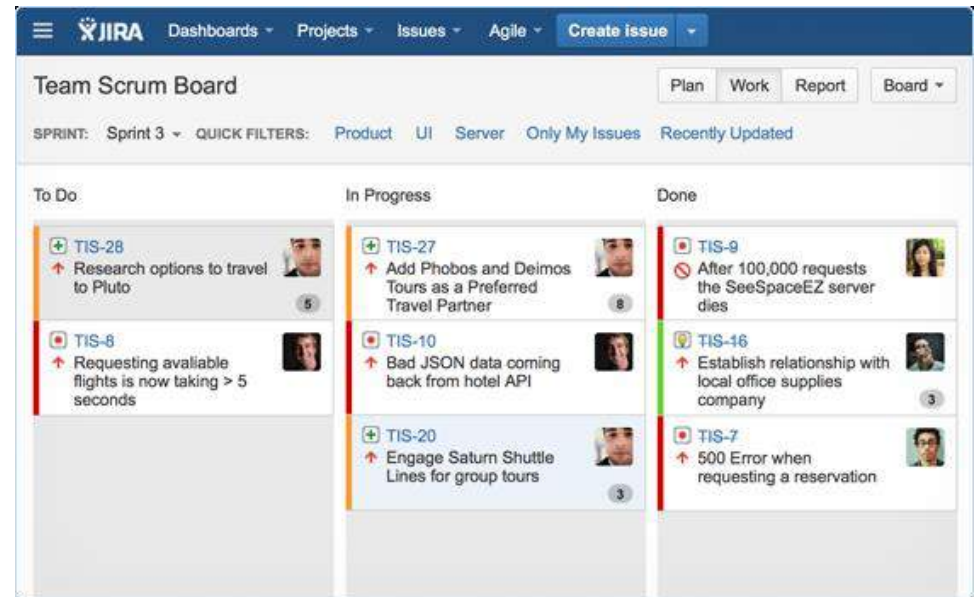
**Regression Cycle Summary:**  
 Started On: 23/Mar/12  
 Build: 766  
 Environment: QA2  
 Progress: 100%

**Final Sanity Tests Cycle Summary:**  
 Started On: 18/Apr/12  
 Build: 766  
 Environment: Staging  
 Progress: 63.16%

# Task Board לוח משימות



ניתן לנהל באופן ידני על לוח אמיתי,  
או ע"י כלים ממוכנים.



מילה יפנית שמשמעותה KanBan  
Visual כרטיס שניתן לראותו )  
שיטה לניהול ושיפור Card  
מתמיד של תהליך העבודה.  
אפשר ליישם ברמת הצוות,  
הפרויקט, ואף פורטפוליו  
הפרויקטים.

# דוגמה ללוח Kanban

The screenshot shows a JIRA Team Scrum Board for 'Sprint 3'. The board is organized into four columns: 'To Do', 'In Progress', 'In Review', and 'Done'. Each column contains task cards with the following details:

Column	Issue ID	Description	Assignee	Count
To Do	TIS-28	Research options to travel to Pluto	[Avatar]	5
	TIS-8	Requesting available flights is now taking > 5 seconds	[Avatar]	
In Progress	TIS-27	Add Phobos and Deimos Tours as a Preferred Travel Partner	[Avatar]	8
	TIS-10	Bad JSON data coming back from hotel API	[Avatar]	
	TIS-25	Engage Jupiter Express for outer solar system travel	[Avatar]	5
In Progress	TIS-20	Engage Saturn Shuttle Lines for group tours	[Avatar]	3
In Review	TIS-58	Add feedback button to the plugin sample code	[Avatar]	
	TIS-45	Email non registered users to sign up with Teams In Space	[Avatar]	2
Done	TIS-9	After 100,000 requests the SeeSpaceEZ server dies	[Avatar]	
	TIS-16	Establish relationship with local office supplies company	[Avatar]	3
	TIS-7	500 Error when requesting a reservation	[Avatar]	
Done	TIS-11	Register with the Mars Ministry of Labor	[Avatar]	2

# פיתוח בספרינטים - Development-

## עדכון סטטוס משימות יומי

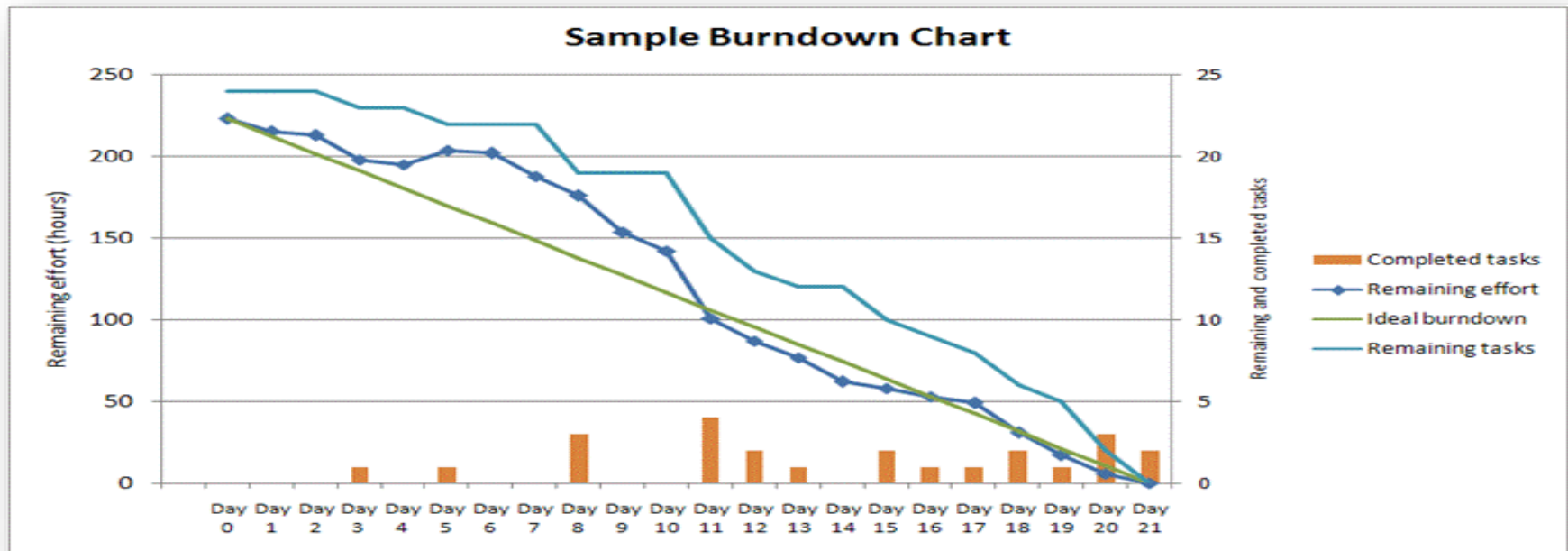
◀ כל איש צוות מעדכן מדי יום את סטטוס המשימות שהוקצו לו לספרינט הנוכחי.

◀ האם המשימה הסתיימה?

◀ אם לא, כמה שעות עבודה נשאר כדי לסיים אותה?

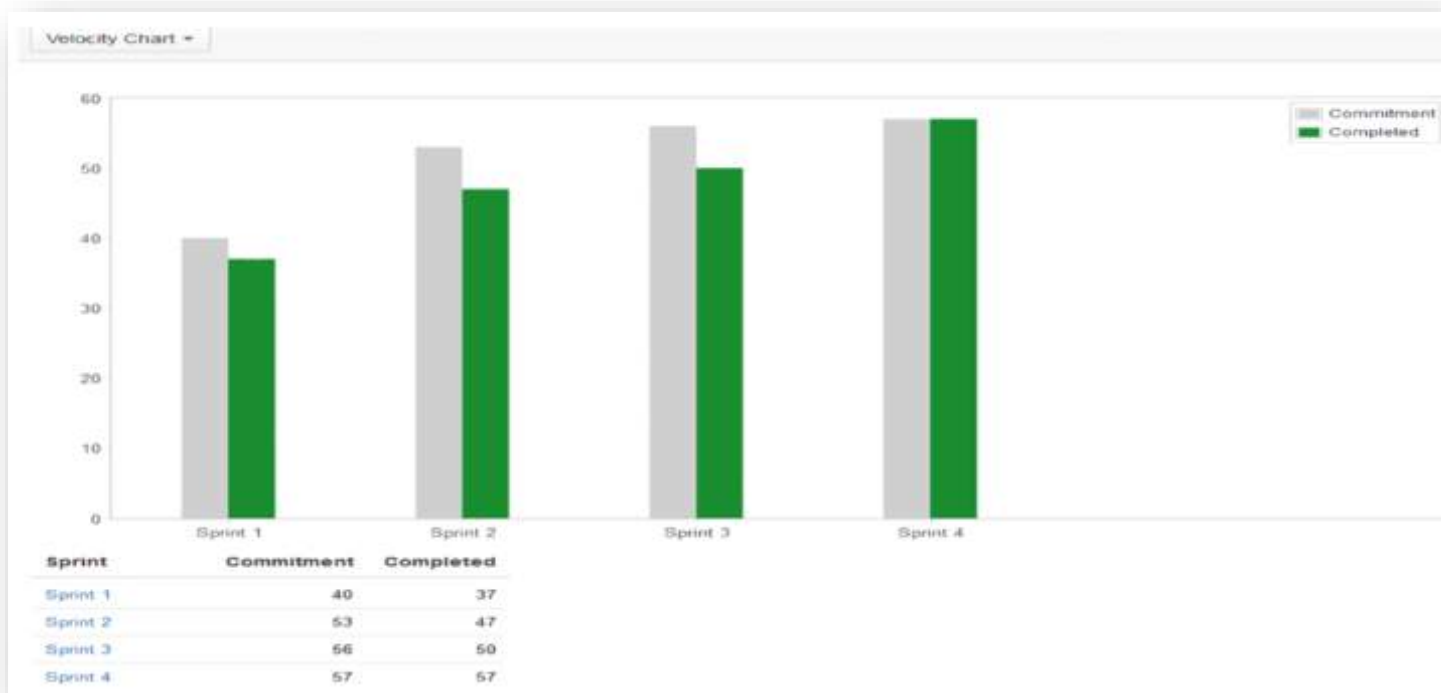
# Burndown Chart

- כלי הבקרה העיקרי של הצוות על התקדמות הספרינט הוא ה-Burndown chart המנטר את כמות העבודה שנותרה בספרינט. התרשים המראה את סה"כ המאמץ שנשאר לבצע בספרינט לאורך זמן. קצב הירידה בגרף מאפשר לחזות את ההסתברות לסיום פיתוח כל תכולת הספרינט עד סוף הספרינט.



# Velocity

- ◀ Velocity מוגדר בתור כמות העבודה שהצוות הצליח לסיים בממוצע בספרינטים האחרונים.
- ◀ מסייע לצוות לתכנן את הספרינטים הבאים ע"י מדידת ה-Velocity של הצוות.





# ביצוע הספרינט - Development

## Daily Meeting (Standup)

◀ פגישות יומיות קצרות לעדכון הצוות. שאלות: מה ביצעתי מאז הפגישה האחרונה? מה אבצע לקראת הפגישה הבאה? מה מונע ממני להשלים המשימות?



◀ בעמידה

◀ 15 דקות

◀ כל משתתף מעדכן בקצרה:

◀ מה הספקתי אתמול?

◀ מה אני מתכוון להספיק היום?

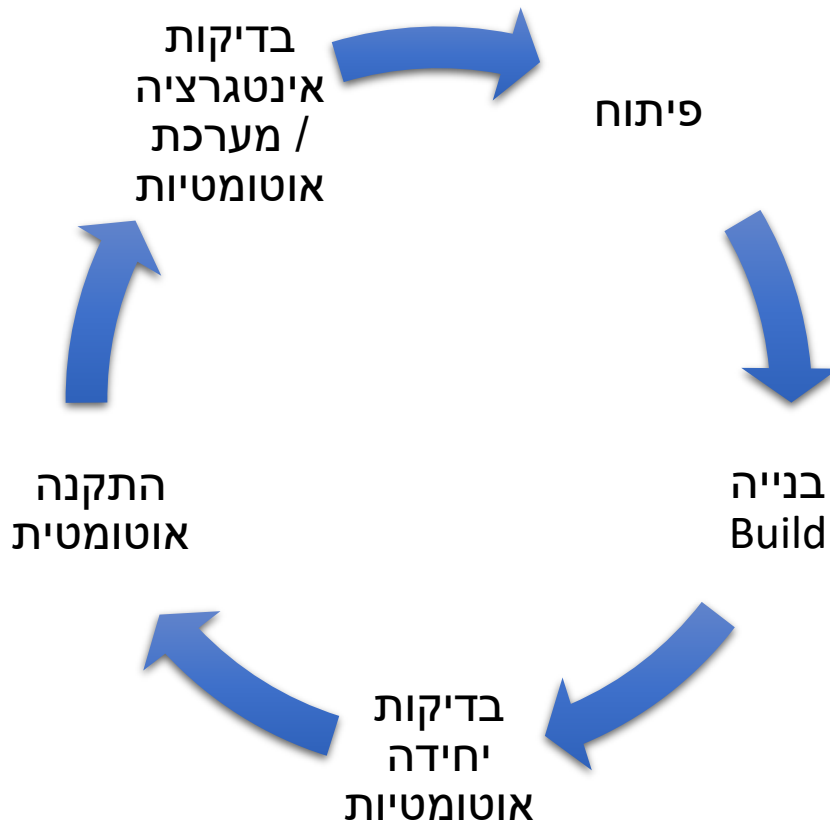
◀ מה מעכב אותי?

◀ לא מטפלים בנושאים פתוחים תוך כדי הפגישה!



# פיתוח בדיקות ואינטגרציה

## Continuous Integration

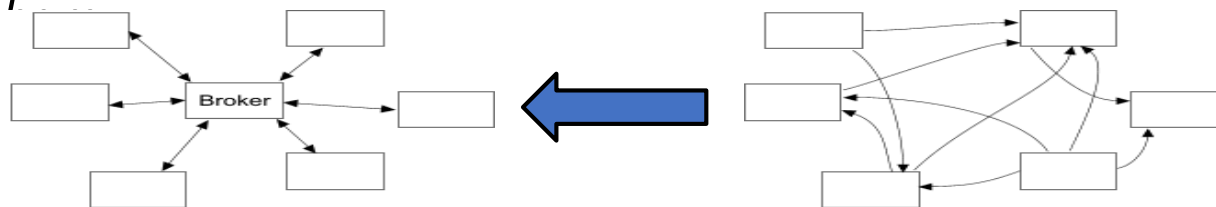


- פיתוח תכולת הספרינט, אינטגרציה ושילוב רכיבים שפותחו עי צוותים אחרים.
- בדיקת מערכת, עומסים ורגרסיה (רצוי כמה שיותר בצורה אוטומטית).
- פיתוח ובדיקת הסבות נתונים (אם רלוונטי).
- פיתוח ובדיקות ואריזת הגרסה.
- עדכון תיעוד נלווה (בעיקר לקראת הספרינטים האחרונים)

# Refactoring

- ◀ שיפור המבנה הפנימי של התוכנה ללא שינוי פונקציונאלי/ התנהגותי.
- ◀ שיפור לצורך הגדלת:
  - ◀ קריאות הקוד
  - ◀ קלות התחזוקה בעתיד
  - ◀ גמישות לשינויים
- ◀ הכרחי לביצוע מדי פעם תוך כדי הפיתוח.
- ◀ יש לשקול הקצאת ספרינט ייעודי רק לביצוע Refactoring.
- ◀ "אי-השלמת Refactoring דומה למינוס בחשבון הבנק: אפשר לחיות עם זה, אבל זה יקר..."

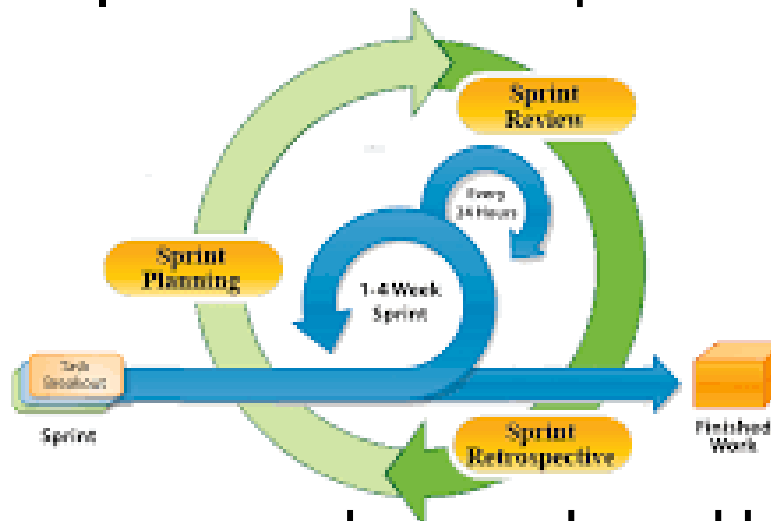
Ward Cunningham



# פגישות סיכום הספרינט

## Sprint Review Meeting

◀ הצגה \ הדגמה למנהל המוצר - Product Owner ולנציגי המשתמשים, בסביבת הבדיקות. קבלת משוב, הערות ובקשות לשינויים ועדכון ה-Backlog.



## Retrospective Meeting

◀ פגישה פנימית של צוות הפיתוח, כולל מנהל המוצר לסקירה והפקת לקחים (מה היה טוב? מה ניתן לעשות טוב יותר?). חשוב לעקוב אחר מימוש של הלקחים בספרינטים הבאים.

ייעוץ, תכנון  
והתארגנות

הקדמה

פיתוח  
ובדיקות

התקנה,  
ייצור ותפעול

# POST GAME ייצור ותפעול

- ◀ בשלב זה קיים יישום עובד בסביבת הבדיקות. ניתן פוטנציאלית להתקינו בסביבת הייצור. העלאה לייצור תתבצע בסיום הספרינט, בסיום הגרסה או בכל מועד שנקבע.
- ◀ התהליך כולל

- ◀ פגישה לאישור התקנה בסביבת הייצור (לפי רשימת תיוג מוגדרת מראש).

- ◀ עלייה לאוויר: התקנה, הסבות נתונים, בדיקות שפיות.

- ◀ ליווי משתמשים בשימוש במערכת וקבלת משוב (בקשות לשינויים, תקלות).

- ◀ אריזת המערכת, התקנה ופעולות לאחר לעליה לאוויר (הסבות נתונים, ליווי משתמשים וכו').

- ◀ ניטור המערכת בייצור.



# DEVOPS

- ◀ מתודולוגית AGILE מאפשרת להוציא גרסאות לעיתים קרובות ולספק עדכונים בקצב גבוה. אולם לאחר שחרור הגרסה, מוטל על גופי התשתית להתקין ולהטמיע אותה.
- ◀ תפיסת ה DEVOPS, נועדה לפרוץ גבולות ומחסומים בין הפיתוח לבין התפעול. מסגרת DEVOPS מרחיבה את מתודולוגיית AGILE גם לשלב ההפצה לייצור ולאורך כל מחזור חיי המערכת.



# העברה לייצור ותפעול

- ◀ תהליכי עלייה לאוויר אוטומטיים ומתמשכים או ידניים כולל התקנה, הסבות נתונים, בדיקות שפיות.

