


# פיתוח אג'ילי



מירב שקרון, [meravgu@gmail.com](mailto:meravgu@gmail.com),  
יוסי זגורי, [yossiza@ariel.ac.il](mailto:yossiza@ariel.ac.il), 052-4668866

# אג'נדה

אג'ייל- מבוא 

Scrum 

XP 

Kanban 

# מתודולוגיות אג'יליות

## • Agile

– אוסף מתודולוגיות פיתוח תוכנה זריזה ואיטרטיבית

– תגובה לכישלון השיטות שקידשו תכנונים במקום

גמישות

– כלי עזר שונים לניהול התהליך

– גישות שונות הניתנות לשילוב



## הגדרה

An iterative and incremental (evolutionary) approach, performed in a highly collaborative manner with just the right amount of ceremony, to produce high quality software in a cost effective and timely manner, which meets the changing needs of its stakeholders.

# הגדרה

An **iterative** and **incremental** (evolutionary) approach, performed in a **highly collaborative** manner with just the **right amount of ceremony**, to produce **high quality** software in a **cost effective** and **timely manner**, which meets the **changing needs** of its stakeholders.

# המניפסט האגילי

## Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.  
Through this work we have come to value:

Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck  
Mike Beedle  
Arie van Bennekum  
Alistair Cockburn  
Ward Cunningham  
Martin Fowler

James Grenning  
Jim Highsmith  
Andrew Hunt  
Ron Jeffries  
Jon Kern  
Brian Marick

Robert C. Martin  
Steve Mellor  
Ken Schwaber  
Jeff Sutherland  
Dave Thomas

# 12 עקרונות אג'יליות

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must work together daily throughout the project.

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design enhances agility.

Simplicity--the art of maximizing the amount of work not done--is essential.

The best architectures, requirements, and designs emerge from self-organizing teams.

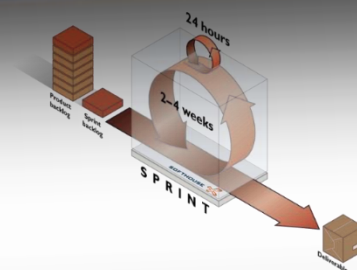
At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

### CHAOS RESOLUTION BY AGILE VERSUS WATERFALL

SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
All Size Projects	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large Size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects	Agile	58%	38%	4%
	Waterfall	44%	45%	11%



# גישת Scrum



# Scrum

- גישת פיתוח המקדמת מיקוד, בהירות ושקיפות לתכנון ומימוש פרוייקטים
- בשימוש ארגונים מגדולים ועד קטנים
- סט ערכים, תפקידים, תהליכים איטרטיבים וכלים
- עבודת צוות, תקשורת בין אישית, העצמת הפרט
- Scrum Master, Product Owner, Developers, Testers
- מחזורי תכנון, תכן, פיתוח ואינטגרציה
- Backlogs, Burn down Charts

# Scrum

- תועלות צפויות מאימוץ Scrum
  - הגברת קצב הפיתוח
  - יישור קו ותאום מטרות הפרט והארגון
  - יצירת תרבות ארגונית מונחית ביצועים
  - תמיכה ביצירת ערך לבעלי העניין
  - יצירת מנגנוני תקשורת וסינכרון יעילים בין כל המעורבים
  - העצמת התפתחות אישית ואיכות חיים חברי הצוות

# Scrum

- <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf#zoom=100>

# Scrum במבט כולל

## axosoft Scrum Diagram:

- 1 Product Backlog**  
The Product Backlog contains a wish list of all the User Stories of a product.
- 2 Release Backlog**  
The goal of a given release is to deliver a subset of the Product Backlog, known as the Release Backlog.
- 3 Sprint Backlogs**  
Each Sprint (or short duration milestone) takes a small chunk of the Release Backlog and gets it Ship-Ready!

### Estimation Techniques:

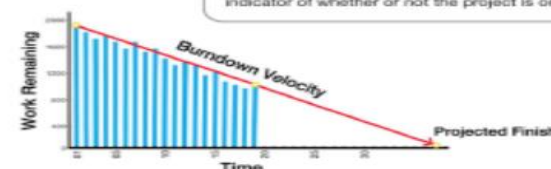
**Hours:** When estimating work, it's important to have the entire team use some standards. Use 1h, 2h, 4h, 8h, etc. No estimations in-between.



**Story Points:** You can also estimate work in comparison to the complexity of a well known but simple component.



Remember that in Scrum, the estimates are only part of the story. The Burndown Velocity is the true indicator of whether or not the project is on track.



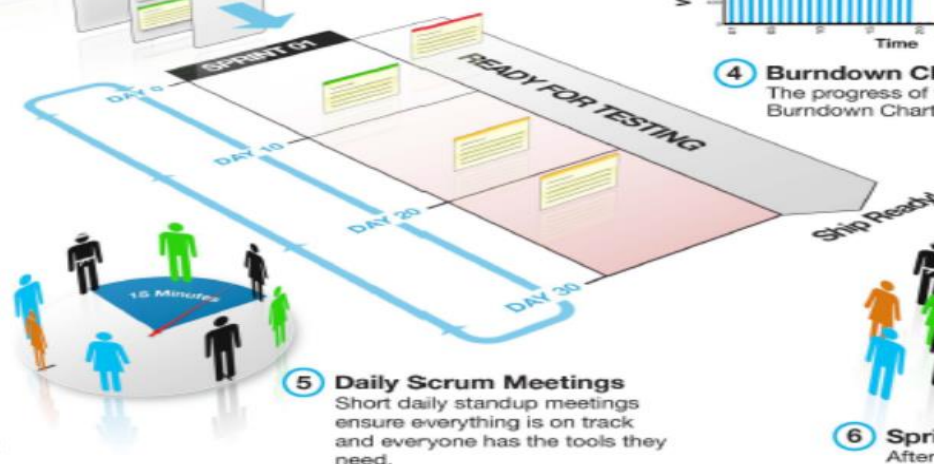
### Team Roles:

- Product Owner:** Is responsible for what goes into the product backlog and prioritizes it. Would probably make a good dictator if given the chance.
- Scrum Master:** A team facilitator. Ensures teams have what they need to get the job done. Also, sets up meetings and monitors everything. Also, kicks ass when necessary.
- Developers & Testers:** They write code and make sure it does what it's suppose to do. Duh!

### For more info:

Intro to Scrum Video: [www.axosoft.com/scrumvideo](http://www.axosoft.com/scrumvideo)

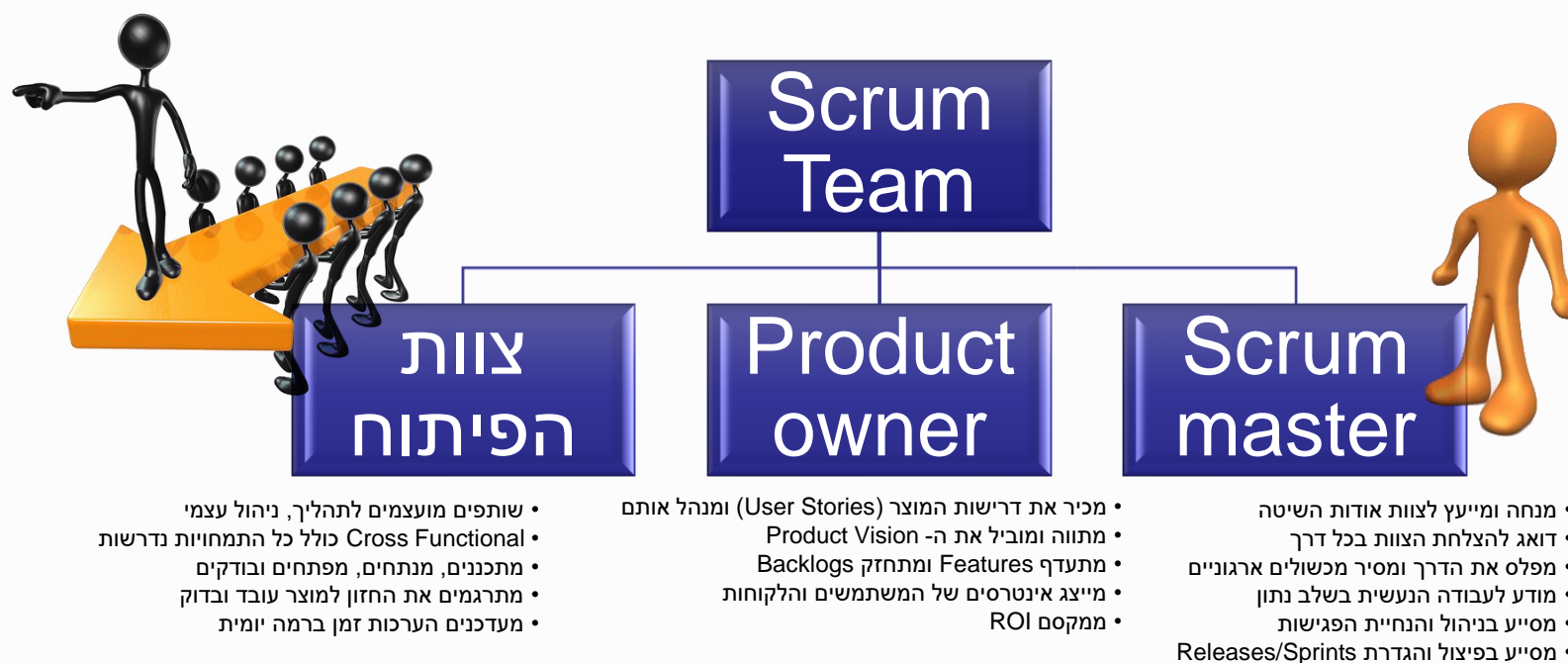
This work is licensed under a Creative Commons Attribution-NonCommercial 3.0 Unported License. © 2012 Axosoft, LLC.



- 4 Burndown Chart**  
The progress of the team is monitored using a Burndown Chart.

- 6 Sprint Retrospective**  
After each sprint, a longer retrospective meeting helps fine-tune the process.

# Scrum - השחקנים



# ומה חסר לנו?



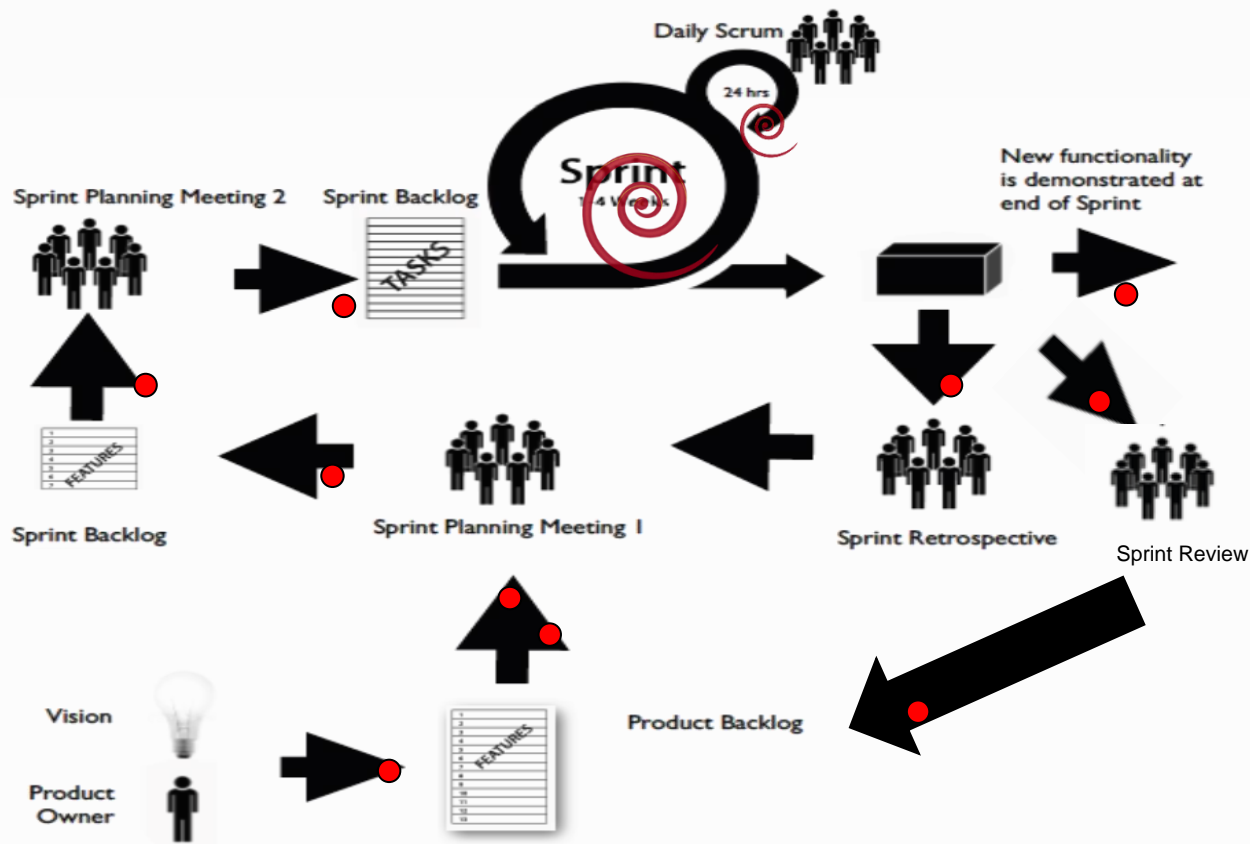
# הנהלה



- אין מנהל פרויקט, הצוות מנהל פעילותו
- תפקיד ההנהלה חשוב אך משתנה
  - יוצרים מודל עסקי המספק משאבים לצרכי הצוות
  - מכבדים את החוקים ורוח השיטה
  - מסייעים בסילוק מכשולים וסוגיות מפריעות
  - מספקים הנחייה ותורמים מניסיונם לצוות
  - מאתגרים את הצוות למצוינות
- מנהל  $\neq$  אומנת, יותר גורל\מנטור\קואוצ'ר
- לא מנחית פתרונות, מתשאל ומכוון לקראת פתרון



# התהליך



# Scrum events

- Sprint Planning •
- Daily Scrum •
- Sprint Review •
- Sprint Retrospective •

# תוצרים ב- Scrum

## SCRUM ARTIFACTS



Product Backlog



Sprint Planning Meeting  
Product Increment



Sprint Backlog



Burndown  
Charts

Daily Scrum  
Meeting



Sprint



Finished Work



Sprint Review  
and Retrospective

# בעיות ב-Scrum

- חוסר בהירות בנוגע לתכולה הסופית של הפרויקט
- "צוות שמנהל את עצמו" – ??
- חלוקת הפיצ'רים ליחידות עבודה קטנות – ??
- אתגר הצוותים הקטנים
- דורש שינויים ארגוניים- ברמת ההנהלה וגם ברמת הצוותים המפתחים
- תפקיד ה-scrum master – תפקיד קשה לביצוע וקשה לאיוש

# השוואה בין מודלים לינאריים ל-Agile

לינארי	Agile
הדרישות בפרויקט	מעט ידוע מראש, הרוב נבנה בהמשך התהליך
שינויים בתכולת הפרויקט	קשיח. גורר עלויות נוספות
רמת הסיכון להצלחת הפרויקט	גמיש. מתקבל בברכה
מעורבות הלקוח בתהליך הפיתוח	גבוהה
מסירה ללקוח	גבוהה מאד בהתחלה, נמוכה בהמשך
אוריינטציה של המודל	באיטרציות
גודל הצוות	לאנשים (ללקוחות ולצוות הפיתוח)
מדדים להצלחה של הפרויקט	צוותים קטנים (3-5)
	הערך העיסקי ללקוח

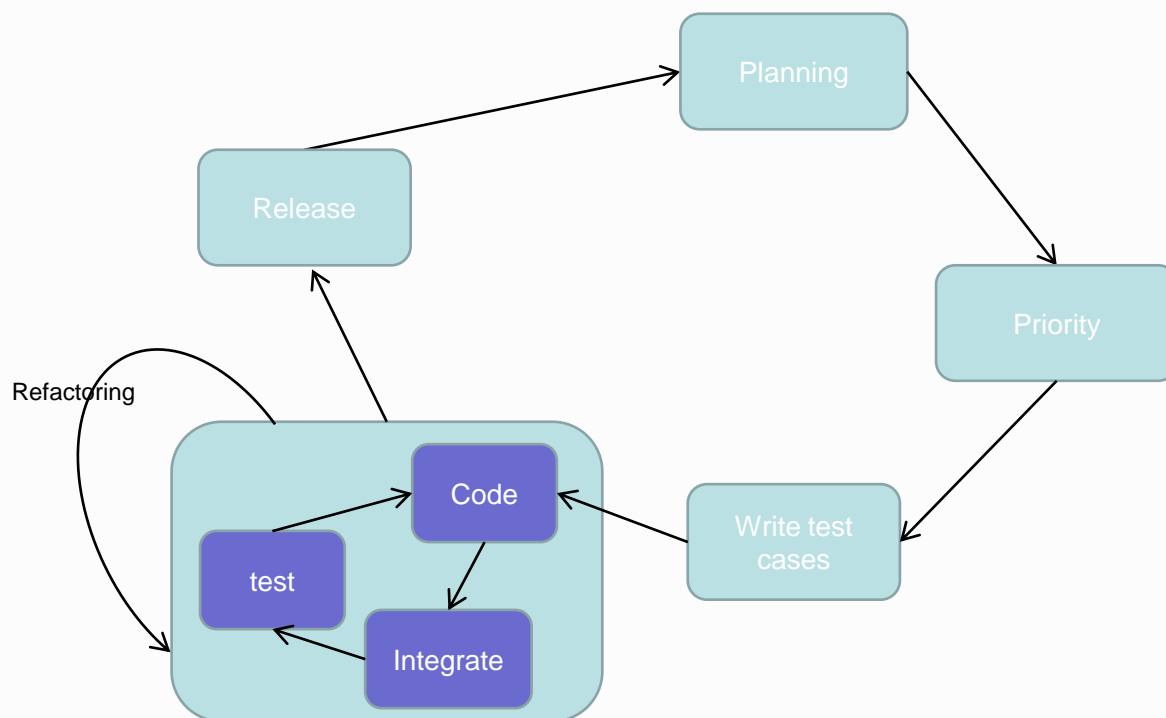
# XP Model

- XP stands for **Extreme Programming**
- טכניקה לתהליך פיתוח תוכנה זריז המבוססת על עקרונות ה-Agile
- נותנת דגש על **מעורבות הלקוח** בתהליך פיתוח התוכנה, תקשורת טובה בתוך הצוותים ועבודה באיטרציות פיתוח.
- המודל פותח ע"י קנת בק בשנת 2000
- הערכים המרכזיים עליהם מושתת XP:
  - תקשורת
  - פשטות
  - משוב
  - אומץ
  - כבוד

# XP של Practices 12

- The Planning Game
- Small Releases
- Metaphor
- Simple Design
- Continuous Testing
- Refactoring
- Pair Programming
- Collective Ownership
- Continuous Integration
- 40-hour week
- On-site Customer
- Coding Standards

# XP Model





# XP Model - יתרונות

- מעורבות הלקוח מגדילה את הסיכוי שהתוכנה המיוצרת יהיה תענה על הצרכים של המשתמשים
- מקטין את הסיכון בפרויקט
- בדיקות ואינטגרציה מתמשכות מסייעות להגביר את איכות העבודה
- זמני הפיתוח קצרים יותר, כי מכינים את הבדיקות בשלב ההגדרות.

# XP Model - חסרונות

- XP מיועד לפרויקט אחד, שמפותח ומתוחזק על ידי צוות אחד.
- XP פגיע במיוחד למפתחים סוליסטים אשר לא עובדים טוב עם אחרים / שחושבים שהם יודעים הכל / שאינם משתפים את הקוד "המעולה" שלהם
- XP לא יעבוד בסביבה שבה לקוח או מנהל מתעקש על מפרט מלא או עיצוב לפני שהם מתחילים לתכנת.
- XP לא יעבוד בסביבה שבה מתכנתים מופרדים מבחינה גיאוגרפית (או לוחות זמנים שונים)

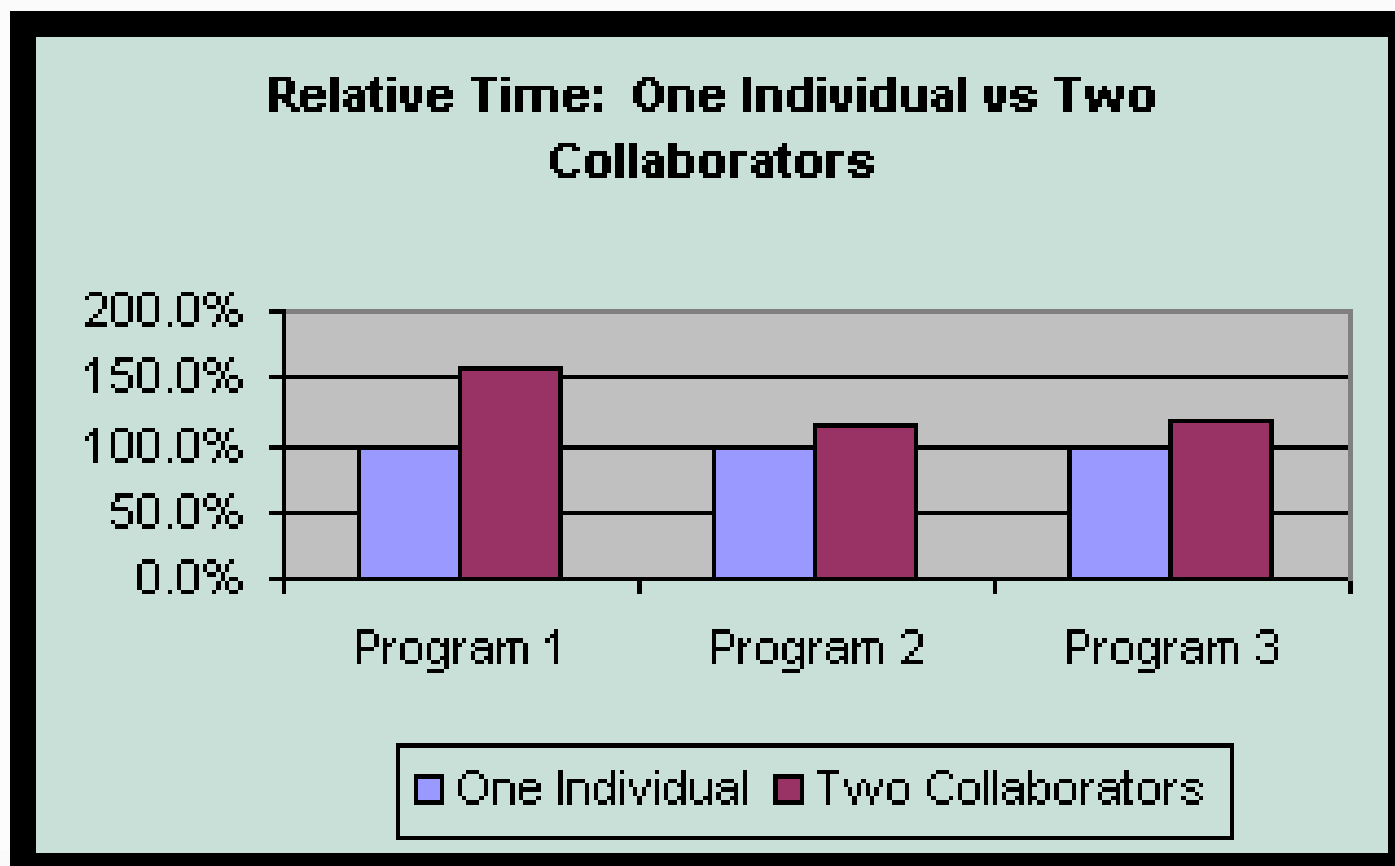
# Pair Programming

- 2 מפתחים יושבים ביחד באותו מחשב
- הזוגות דינאמיים
- רעיון הוא ששני אנשים בוודאות יכתבו קוד יותר טוב ויותר יעיל ממישהו אחד (ועם פחות באגים)
- 2 המתכנתים הם בעלי רמות שונות של ניסיון, אבל זה לא נועד לחניכה של מפתח חדש



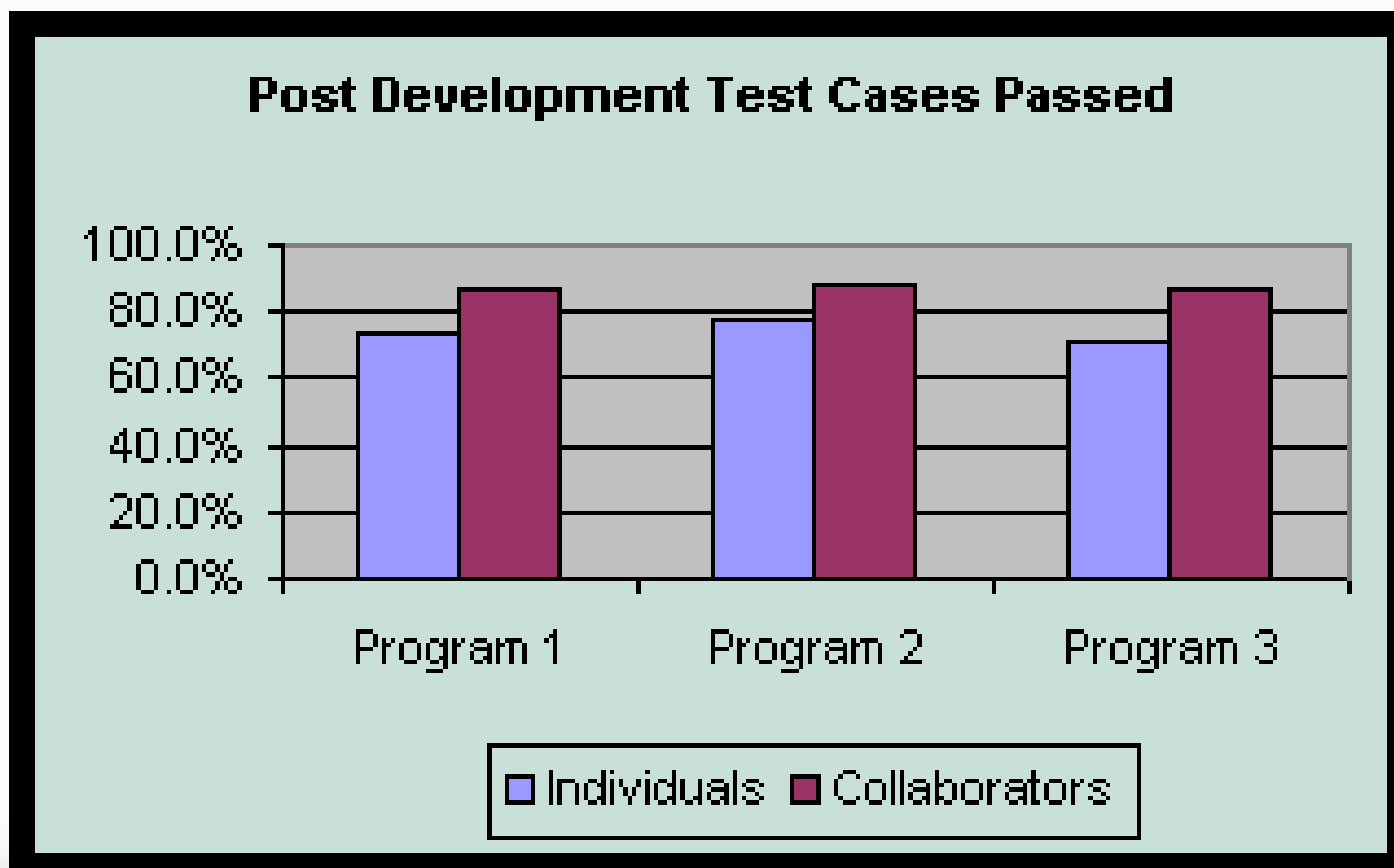
# אז זה טוב?

University of Utah Experiment: Pairs spent 15% more time on the program than individuals



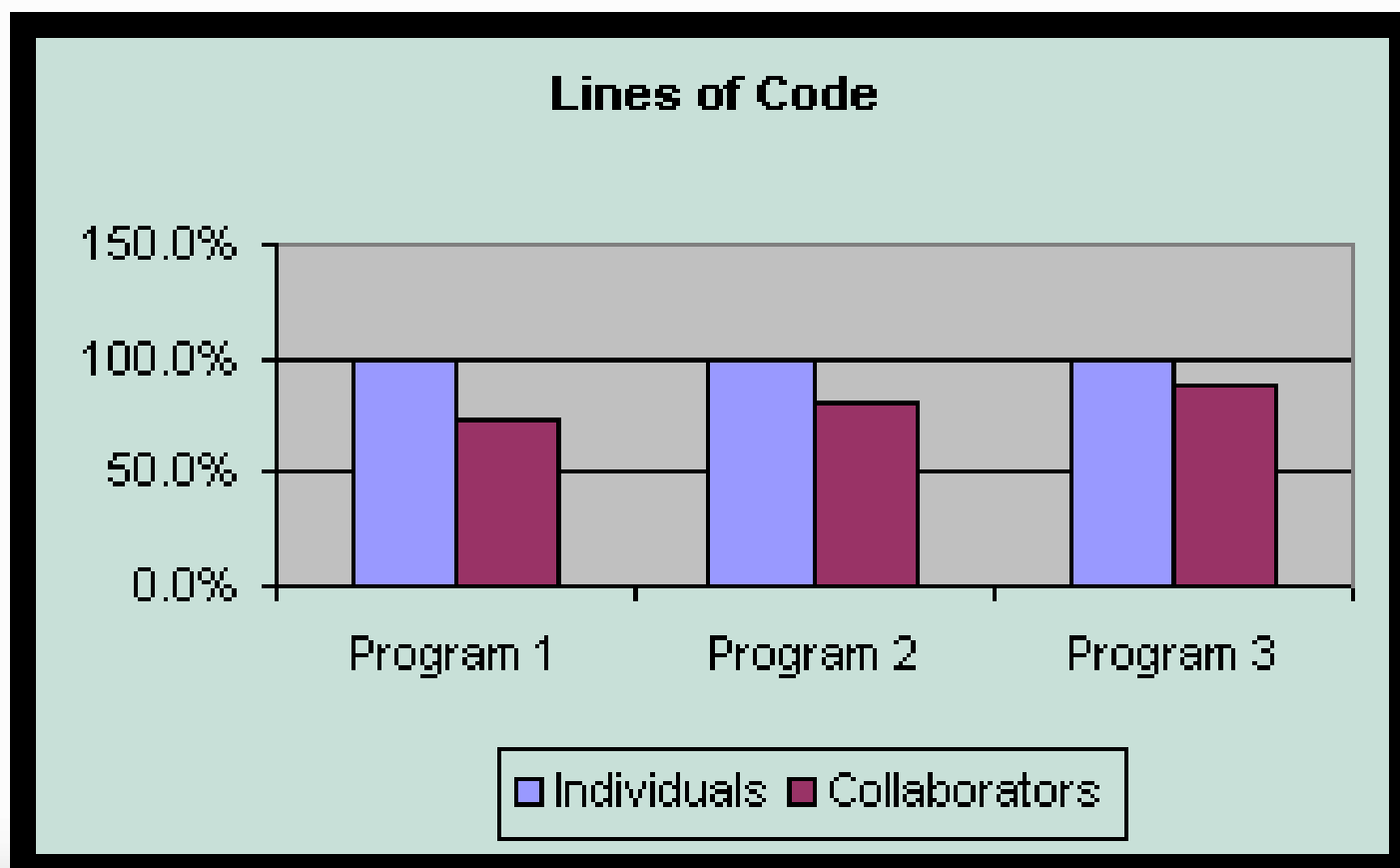
# אז זה טוב?

University of Utah Experiment: Code written by pairs passed more test cases than code written by individuals



# אז זה טוב?

University of Utah Experiment: Pairs consistently implemented the same functionality produced by individuals in fewer lines of code



# אין מסקנה חד משמעית

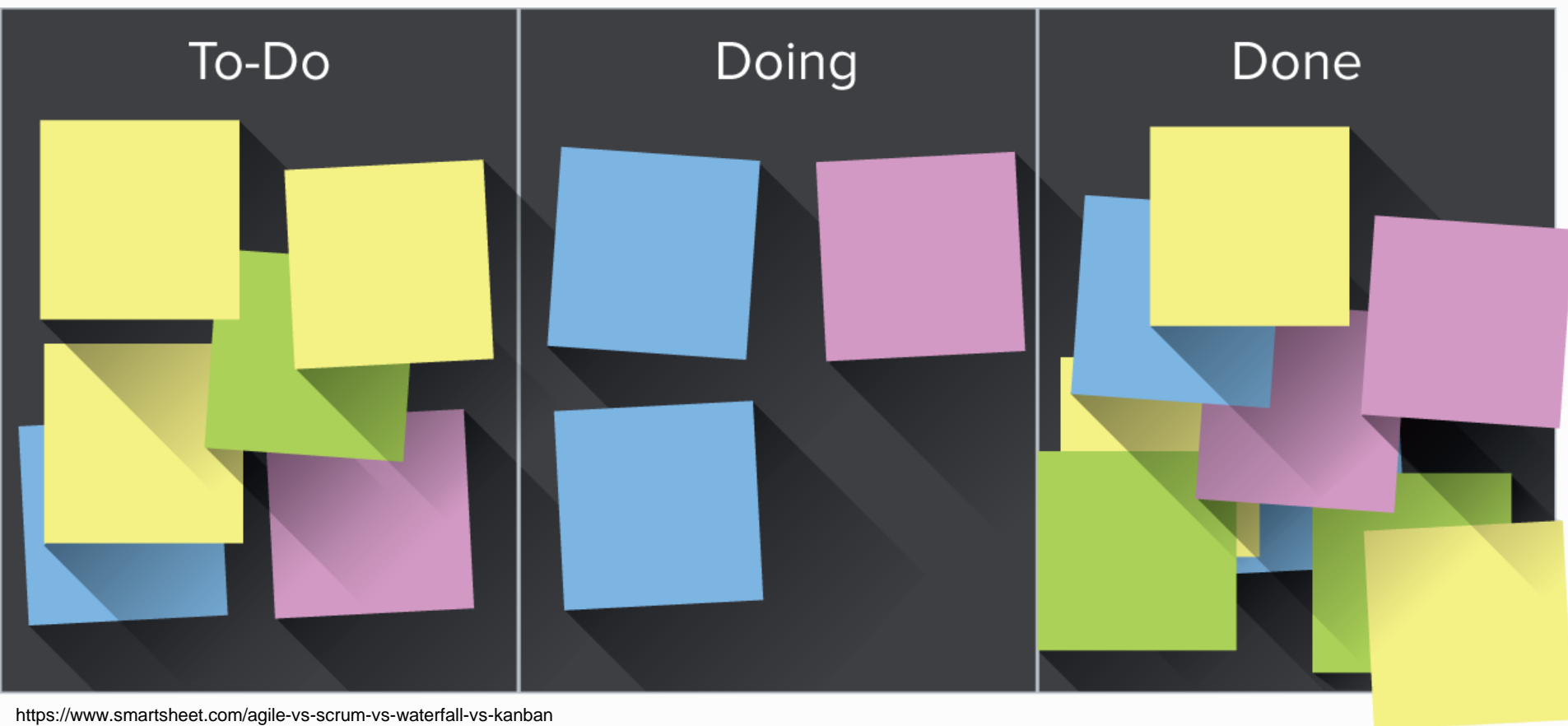
- ישנם מחקרים שמראים שפיתוח בזוגות הוא מאד מוצלח כמו-
  - מה שראינו קודם
- Williams, Laurie, et al. "Strengthening the case for pair programming." *IEEE software* 17.4 (2000): 19-25.
- וישנם מחקרים שמראים את ההיפך, או לפחות לא מצליחים להוכיח שיפור משמעותי
  - Nawrocki, Jerzy, and Adam Wojciechowski. "Experimental evaluation of pair programming." *European Software Control and Metrics (Escom)* (2001): 99-101.

# model Kanban

- Kanban ביפנית זה כרטיס או "סימן ויזואלי"
- טכניקה לתהליך פיתוח תוכנה זריז המבוססת על עקרונות ה-Agile
- Kanban נותנת דגש לצד הוויזואלי של התהליך – מה לייצר, מתי לייצר וכמה לייצר.
- השיטה מעודדת ביצוע שינויים קטנים והדרגתיים במערכת.



# Kanban board



<https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban>

Kitchen Remodeling | Trello

https://trello.com/board/kitchen-remodeling/50046064994af80a340bc96b


Trello

Notifications Boards

### Kitchen Remodeling

#### Ideas

install pot rack over island



replace drawer knobs with quirky ones from antique shop

1

Twitter refrigerator integration

new window valence to match cabinet colors

new coat of paint for door frame

Add a card...

#### To Do

Adjust water pressure in sink

0/4

remove old refrigerator and stove

install new sink

1 Jul 23 0/9

install new refrigerator and stove

install new wall shelves

1

install new flooring


Buy paint for cabinets

Add a card...

#### Doing

Pick faucet to match new sink

Pick countertop colors



Buy new kitchen cart

1

Add a card...

#### Done

Design new kitchen space

1 vote 2

call contractor

Add a card...

## השוואה של עקרי תפיסות העולם

Scrum	Kanban	
לכל חבר צוות יש תפקיד מוגדר מראש, שבו Scrum Master משמש כמאמן אג'ייל של הצוות, Product Owner מגדיר מטרות ויעדים וחברי הצוות מבצעים את העבודה תוך עבודה צמודה עם ה-Product Owner	"אין תפקידים מוגדרים מראש עבור צוות. מעודדים את הצוות לתקשר ולפעול ביחד כאשר אחד מאנשי הצוות תקוע או עמוס בעבודה	תפקידים ואחריות
זמני אספקת תוכנה נקבעים באמצעות הגדרת ספרינטים	מוצרים ומשימות מסופקים באופן רציף על בסיס צורך (on demand)	מועדי אספקה של תוכנה עובדת
יש שימוש ב"משיכת משימות", אבל ברמה של קבוצה של משימות הנמשכת כולה בכל ספרינט (קבוצת המשימות נקבעת בתכנון הספרינט)	חברי הצוות "מושכים" משימות חדשות לאחר שהמשימה הקודמת הושלמה ובתנאי שכמות המשימות בשלב הנוכחי קטן מהחסם העליון שהוגדר עבור שלב זה	טיפול במשימות
מדידת תפוקה באמצעות ההספק הממוצע בספרינטים הקודמים. ההספק נמדד בדרך כלל על ידי מושג הנקרא Story Points או על ידי מספר המשימות שהסתיימו בספרינט	מדידת תפוקה באמצעות "זמן מחזור" (הזמן העובר מתחילת טיפול במשימה עד סיום הטיפול המשימה)	מדידת תפוקה
שינויים במהלך ספרינט אינם רצויים. הספרינט הוא "אזור זמן מוגן" עבור הצוות	ניתן לבצע שינויים בכל נקודה בזמן	שינויים

# Kanban model

## יתרונות:

- מגביר את הגמישות לשינויים
- מצמצם ביזבוז זמן- תמיד ממתינה למתכנת עוד משימה
- קלה להבנה ולהטמעה, ניתן להשתמש על גבי מתודולוגיה אחרת
- מקצר את זמני המסירה של משימות

## חסרונות:

- כל הזמן צריך לעדכן את הלוח
- אין בלוח מידע לגבי זמני המסירה

# אז מה לבחור?

• תלוי!!!

• יש להתחשב בגורמים הבאים:

- הארגון- מטרות, אילוצים, תשתיות...
- ההנהלה- פתיחות, הבנה מקצועית, רצון לשינויים...
- העובדים- הכשרה מקצועית, יחסי אנוש, פתיחות...
- המוצר- מורכבות, ייחודיות, האם הדרישות ידועות...

# הסכם שלום בעולם האג'ייל? Kanban ו-Scrum הוציאו מדריך משותף

כתב אורח | בתאריך 28 במאי 2018 אין תגובות פיתוח

שתף/שתפי ציוץ

באחרונה יצאו שני המחנות בהצהרה משותפת: במקום לבחור, אפשר וצריך לשלב על מנת ליצור שיטה שלמה ומועילה יותר. מה ההבדל בין שתי השיטות ומה מציע המסמך החדש?



<https://www.geektime.co.il/kanban-scrum-peace-agreement/>