

הנדסת תוכנה

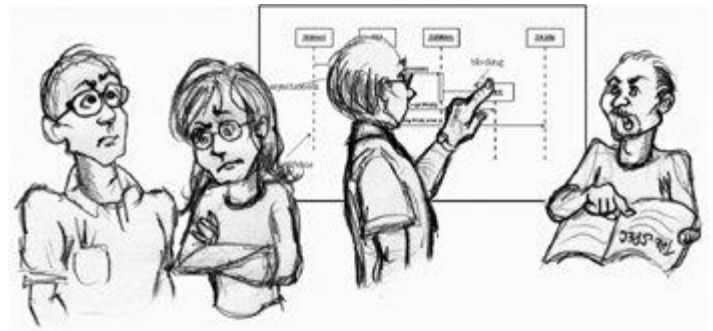
10. קוד קיים – Legacy Code

(מבוא לעקרונות תיכון ותבניות מונחי
עצמים - OODP)

“Simplicity is prerequisite for reliability”
- E. W. Dijkstra

"Writing code a computer can understand
is science.

Writing code other programmers can
understand is an art.”, [Jason Gorman](#)



מה היום?

- ראינו: בדיקות ברמות שונות כדרך למוצר איכותי
- בהמשך עוד על **איכות תוכנה**: עקרונות תיכון מונחה עצמים
- Object Oriented Design Principles, תבניות תיכון
- משימת סבב 3 (אחד לפני אחרון) - סקר שיפורי קוד
- משימה אישית 5 (רשות) – Refactoring?
- היום: מבוא: עבודה עם קוד קיים
- הרצאה 3\תרגיל:
 - מצגות סבב
 - סקר בדיקות יחידה

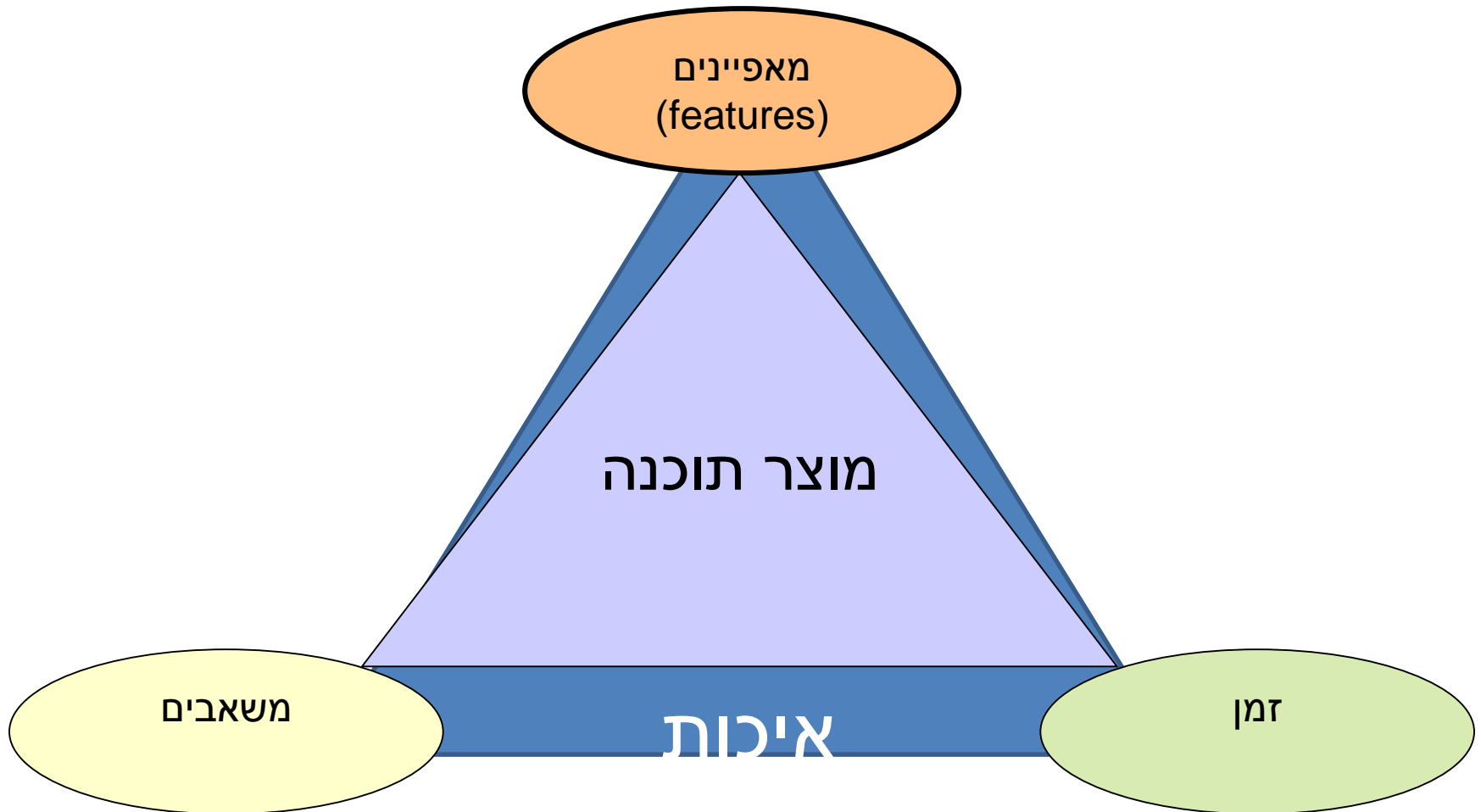
מקורות

- Feathers, Working Effectively with Legacy Code
- Fowler, Refactoring
- Berkeley/Edx SAAS course

קישורים להרחבה

- [How about learning f***ing programming?](#)
Short video on testing/refactoring/design
- Fowler, [An Appropriate Use of Metrics](#), 2013
- Smell and Refactoring lists:
 - <http://www.soberit.hut.fi/mmantyla/BadCodeSmellsTaxonomy.htm>
 - <http://www.codinghorror.com/blog/2006/05/code-smells.html>
 - <http://users.csc.calpoly.edu/~jdalbey/305/Lectures/SmellsToRefactorings>

תזכורת: פרויקט תוכנה:



איכות תוכנה

- מרכיבים פנימיים

- נראים למפתחים

- דוגמאות:

- מודולריות

- עקיבות

- פשטות \ קריאות

- סגנון הקוד

- יכולת להשתנות ולהתפתח

[Begel & Simon 08], עובדים חדשים
במיקרוסופט מבלים את רוב השנה
הראשונה בקריאת קוד

- מרכיבים חיצוניים:

- נראים ללקוח\למשתמש

- דוגמאות:

- בעלי ערך ללקוח! (TTM)

- נכונות, יעילות

- שמישות

- עמידות

- הרחבתיות?

➤ אי אפשר את כולם ביחד

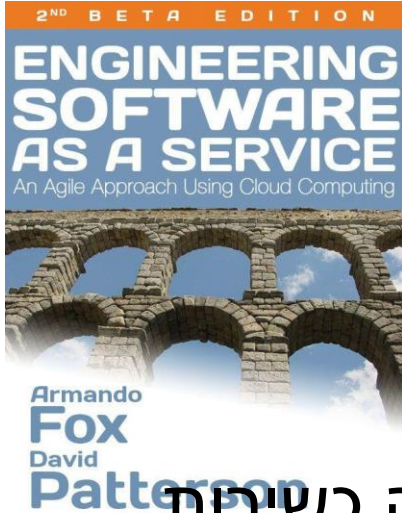
Butler W. Lampson, [Hints for Computer System Design](#), 1983
(2013 [lecture](#))



איכות תוכנה

- איך משיגים תוכנה איכותית?
- בתיכון ראינו יעדים כלליים: צימוד (coupling) נמוך, לכידות (cohesion) גבוהה כיצד משיגים אותם?
- עקרונות + תבניות + הרגלים = תוכנה איכותית
- Beck's Simple Design
- פיתוח תוכנה מונחה עצמים שולט (?), לכן נדון מכיוון זה בהמשך

מתוך קורס הנדסת תוכנה בברקלי



- קורס ותיק ([וידאו](#), [2013](#))
Fox & Patterson –
peer instruction –
Industry: legacy code –
(obamacare) –
- קורס [מקוון](#) ([I](#) [II](#) coursera/edx) בשילוב תוכנה כשירות (SaaS) < K100 סטודנטים + [ספר](#)
- [קורס](#) בחירה דומה במכללה (הנדסת תוכנה למחשוב ענן)
 - נלמד הפעם כמה עקרונות כלליים יותר
 - לפי הזמן גם מונחה עצמים
 - גיוון והשוואה

שקפים

- <https://drive.google.com/#folders/0BxKnkWLvWBrTVG9tQUk2a1p2eFk>
– 08 - Legacy



What Makes Code “Legacy”
and How Can Agile Help?
(ESaaS §9.1)
Armando Fox



Wrapup of Refactoring
example

לסיכום

- קוד קיים \ Legacy Code
- Refactoring \ שיפרוק?
- תיכון ואיכות מתמשכים, הקשר לבדיקות
- האם הקוד שלכם כבר legacy?
- משימת סבב לפני אחרון – סקר שיפורי קוד
- משימה אישית 5 (רשות):
- Fork: <https://github.com/wouterla/GildedRose>
- Submit a well refactored code (with commits)
- תאריך: הרצאה אחרונה

בפעם הבאה - ?

- המשך עקרונות תיכון מונחה עצמים
- מימוש מקובל של עקרונות:
- תבניות עיצוב (תיכון) Design Patterns
- עוד על Refactoring
- אורח?
- עוד: חווית משתמש, כלים 2