

# 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](#)



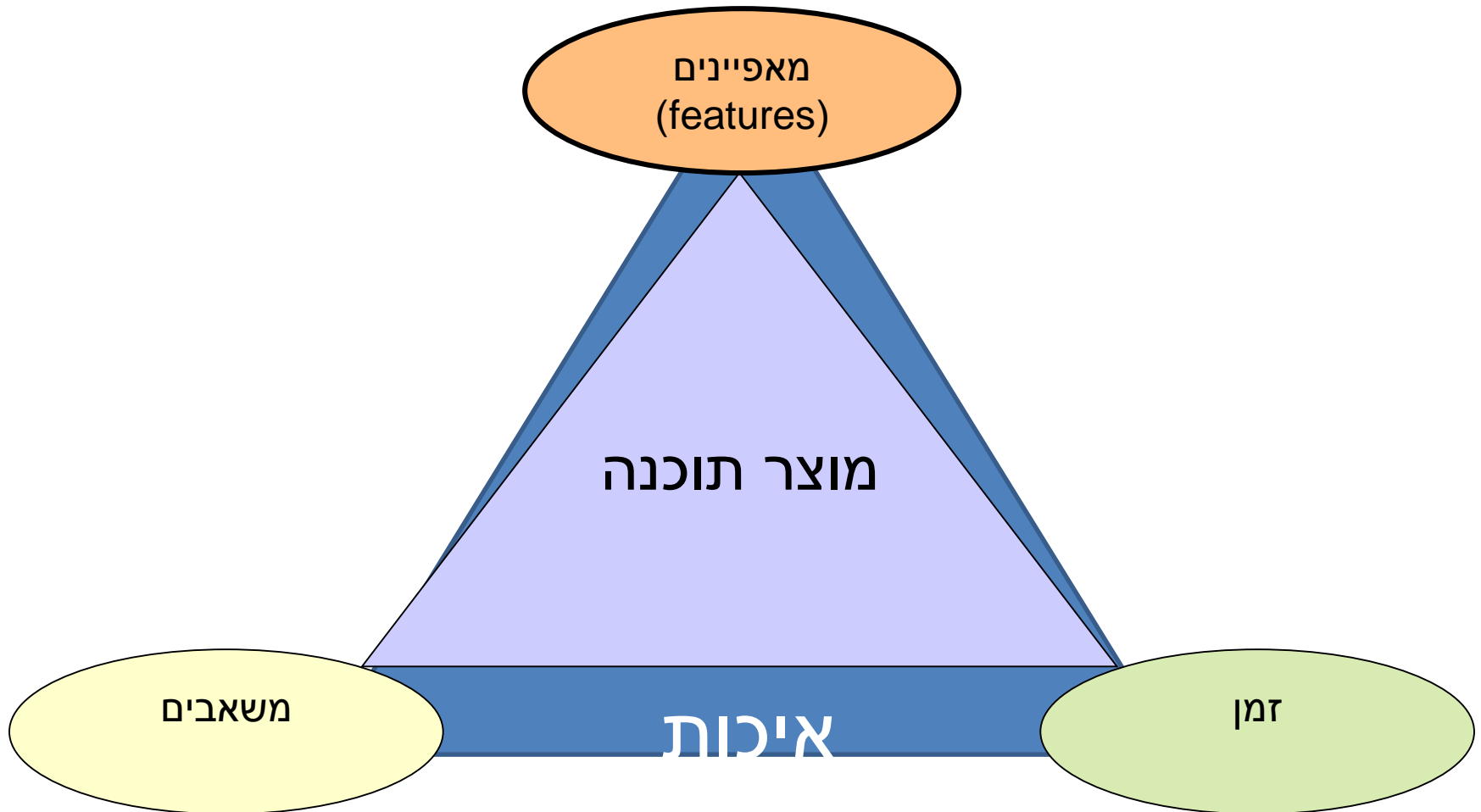
# מה היום?

- 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)
  - משימה אישית 4 (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

# תזכורת: פרויקט תוכנה:



# איכות תוכנה

- מרכיבים חיצוניים:
  - נראים ללקוח\למשתמש
  - דוגמאות:
    - בעלי ערך ללקוח!
    - שמישות
    - נכונות
    - עמידות
    - הרחבתיות?
- מרכיבים פנימיים
  - נראים למפתחים
  - דוגמאות:
    - הסתרת מידע
    - עקיבות
    - פשטות \ קריאות
    - סגנון הקוד
    - יכולת להשתנות

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

# איכות תוכנה

- איך משיגים תוכנה איכותית?
- ראינו יעדים כלליים: צימוד (coupling) נמוך, לכידות (cohesion) גבוהה  
כיצד משיגים אותם?
- $\text{עקרונות} + \text{תבניות} + \text{הרגלים} = \text{תוכנה איכותית}$
- פיתוח תוכנה מונחה עצמים שולט (?), לכן נדון מכיוון זה בהמשך

# מתוך קורס הנדסת תוכנה בברקלי

- קורס בן כמה שנים מקביל לשלנו ([וידאו](#), [2013](#))
  - Fox & Patterson
  - peer instruction
  - Industry: legacy code
- לאחרונה כקורס [מקוון](#) ([I](#) [II](#) coursera/edx) בשילוב תוכנה כשירות (SaaS) < K100 סטודנטים + [פּר](#)
- [קורס](#) בחירה דומה במכללה (ענן וה"ת II)
  - 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
- האם הקוד שלכם כבר legacy?
- משימת סבב לפני אחרון – סקר שיפורי קוד
- משימה אישית 4 (רשות):
- Fork: <https://github.com/wouterla/GildedRose>
- Submit a well refactored code (with commits)
- תאריך: הרצאה אחרונה

# בפעם הבאה - ?

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