

# הנדסת תוכנה

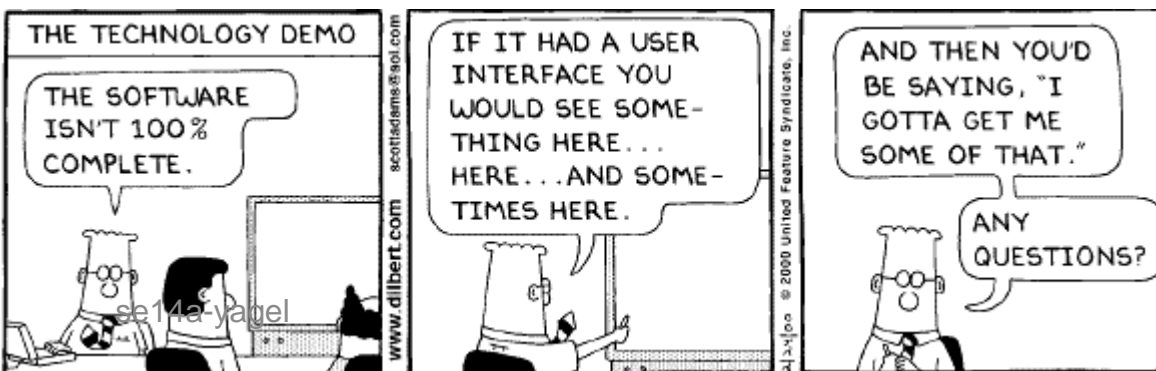
## 13. סיכום הקורס ומצגות סיום



Pragmatic Programmer Tip :  
**learn a new language every year**

# מה היום?

- סיכום הקורס
  - מה הלאה?
  - על המבחן
- "מסיבת שחרור" – מצגות (CD+) ושיקוף
- ~~תרגיל: השלמות ובירורים~~



# לסיכום הקורס

- האם בכלל צריך כזה קורס? האם אפקטיבי
- מיקום בתואר וקשר לקורסים אחרים
- האם וכמה תרם הפרויקט? ההרצאות?
- גודל קבוצות? סוג משימה?
- מה מיותר\מעמיס? מה חסר?
- התנסות מול "less is more"
- סקר...
- מי שבכל זאת אהב (או רוצה לשנות) מוזמן להצטרף לצוות הקורס...

# כמה שקפים מהרצאת המבוא

# מהי הנדסת תוכנה?

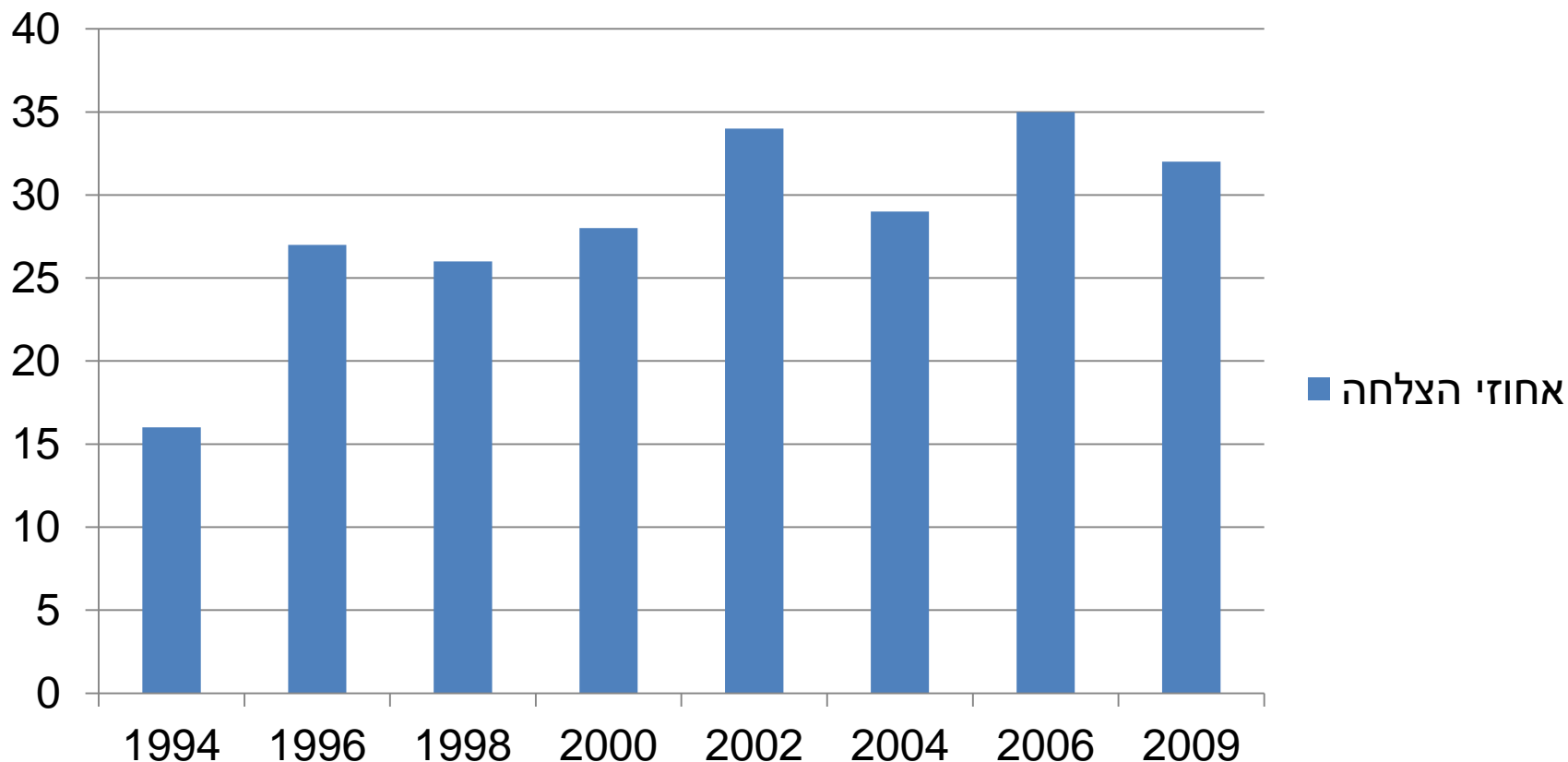
**“Software Engineering** is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, and the study of these approaches; that is, the application of engineering to software”

[IEEE SWEBOK’04], Wikipedia

Software engineering has accepted as its charter,  
"How to program if you cannot." -- *E. Dijkstra*

# Chaos Report (debatable...)

## אחוזי הצלחה של פרויקטי תוכנה



# מדוע פרויקטי תוכנה נכשלים לעיתים כל כך קרובות? \*

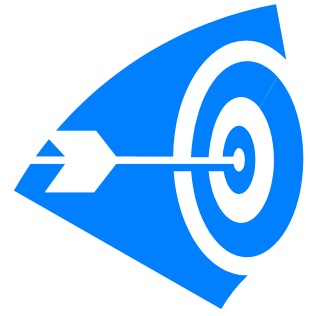


- יעדי הפרויקט לא מציאותיים או לא ברורים
- אומדנים לא מדויקים של המשאבים הנדרשים
- דרישות מערכת אינן מוגדרות היטב
- דיווח לקוי לגבי מצב הפרויקט
- סיכונים לא מנוהלים
- תקשורת לקויה בין הלקוח, המפתחים והמשתמשים
- שימוש בטכנולוגיה לא בשלה
- אי-יכולת לנהל את מורכבות הפרויקט
- פרקטיקות פיתוח מרושלות
- ניהול לקוי של הפרויקט
- פוליטיקה של בעלי עניין
- לחצים מסחריים

בסיום הקורס נחזור לרשימה  
זו ונבדוק האם רכשנו כלים  
מתאימים להתמודד עם  
בעיות אלה.

\* [Charette, R. N., Why Software Fails?, IEEE Spectrum, Vol. 42, Issue 9, Sept. 2005](#)

# מטרות הקורס (סילבוס)



- הבנת הבעיות והפתרונות המרכזיים של הנדסת תוכנה בפיתוח מוצרי תוכנה.
- פיתוח ראיה מערכתית והיכרות עם תהליכים, שיטות עבודה וכלים רלוונטיים בשימוש התעשייה.
- לימוד והתנסות בבניית פרויקט תוכנה משמעותי תוך כדי עבודת צוות וכישורים רכים נוספים.
- הכנה לפרויקט הגמר.



# הפרויקט - מטרות

- נסיון ישיר עם חומר הקורס
- אתגרים טכניים בשל גודל הפרויקט
- אתגריים חברתיים במסגרת מאמץ קבוצתי
- הזדמנות להתנסות בסביבה חדשה, למשל: התקנים ניידים, Web, .net, Ruby, APIs, קוד פתוח, רשתות חברתיות, דרייברים, תוסף דפדפן, [html5](#) ....  
(בכל זאת מומלץ RAD)
- הזדמנות עסקית (זכויות יוצרים!)
- **המלצה: לא לפתוח הרבה חזיתות!**
- **מה בכל זאת שונה מהתעשייה?**



# לסיכום

- דיון בהנדסת תוכנה, תוכנה היום, אתגרים
- מטרת הקורס: מתכנת  $\leq$  מהנדס תוכנה
- המיוחד בקורס
  - רב תחומי
  - הזדמנות לעבוד על רעיון שלכם
  - בד"כ אין תשובה אחת נכונה, מותר לטעות
  - כישורים "רכים" (יצירתיות, שיתוף פעולה)
  - לא קשה, אבל עבודה די רבה (ומהתחלה!)
- תוכן הקורס: תהליכים, דרישות, תיכון, בדיקות, מימוש, כלים ועוד.....
- שאלות \ הבהרות \ הצעות ?

## בהצלחה ובהנאה



# מה למדנו: תהליכים ושיטות

- מפל המים, פיתוח איטרטיבי, אג'יל, אתחול פרויקט, ניהול, עבודת צוות, דרישות, תיעוד, ניתוח, הערכה, תכנון, תיעדוף, רטרוספקטיבה, תיכון (מונחה עצמים), ארכיטקטורה, בדיקות, חווית משתמש (שמישות), איכות, סקרים (עמידה לפני קהל)

<http://scottberkun.com/2011/scrum-for-weddings/>

- Version Control, **TDD**, OODP, Refactoring/Reuse/Legacy Code

# Computer Aided Software Eng.

- Collaboration / Project: Wiki, Issue/Bug Tracking, Planning, **VCS (git / github)**, Code Review, etc.

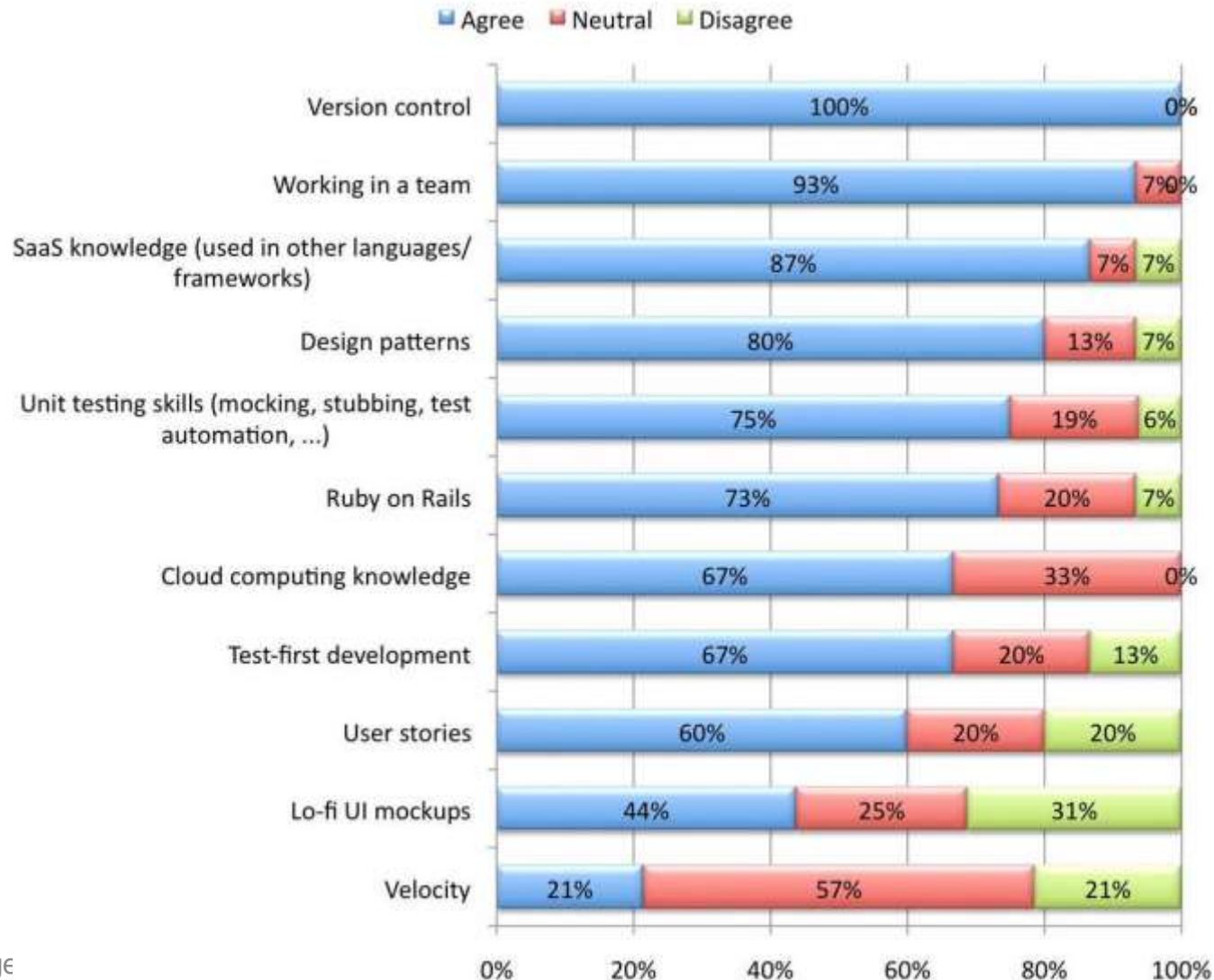


SCISR9\_(mp3cut.net34-36).mp3

[resume](#) ([StartupEng. ~1:40](#), [Oredev Conf. ~21](#))

- Test & Design Tools: Unit Testing, Mocking Framework, Coverage, CI, Refactoring, UML TOOLS, Measurements
- Standard Development & Productivity Tools

# סקר בוגרים בברקלי, אלו מנושאי קורס דומה הכי חשובים

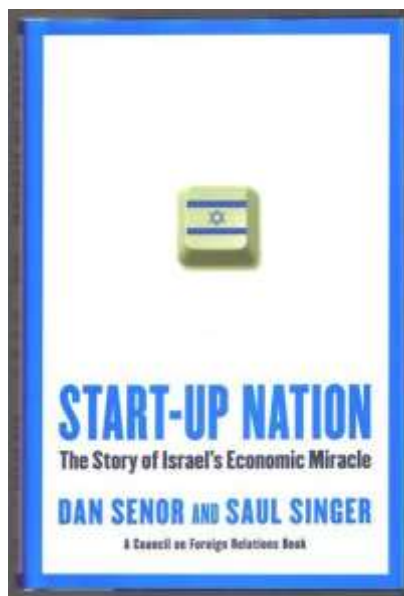


# מה אולי חסר (לדעתי)

- מידת תיאום הפרויקט לקורס
- עוד כלים... (תיעוד, ביצועים, דיבאג, ניהול משימות, למשל [Huboard](#) , [PivotalTracker](#))
- עוד תהליכים ושיטות (סקרים\לקוחות, הפחתת סיכונים, מדידת התקדמות\velocity, הערכת עמיתים, תחזוקה)
- גיבוש צוות
- שיווק, פטנטים, סטארט-אפ
- המשך בקורס בחירה \ תואר שני

# מקומה של התוכנה

	IN THE OLD DAYS	NOW
LISTENING TO MUSIC		
WATCHING FILMS		
CHATTING WITH FRIENDS		
READING THE NEWS		
PLAYING MUSIC		



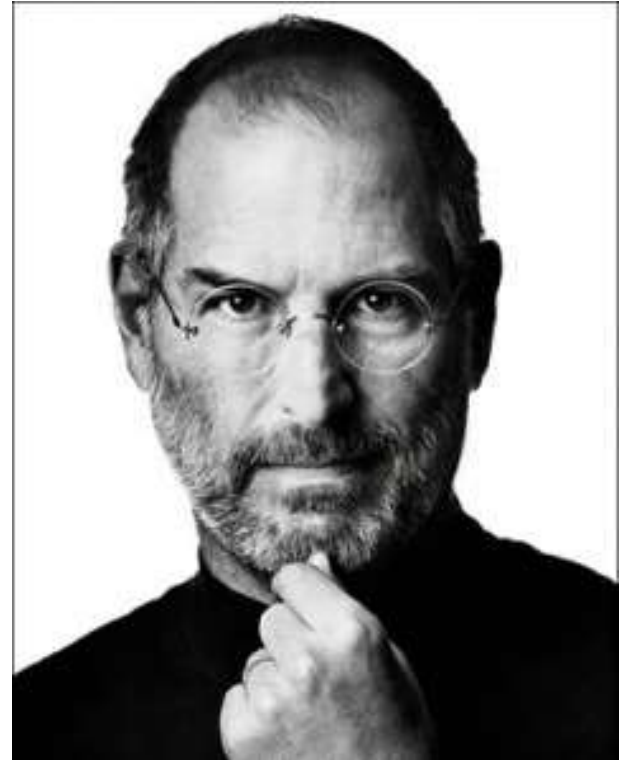
# איך להישאר מהנדס תוכנה רלוונטי

- ניסיון מעשי
- חונך טוב
- לימוד מתמיד: קריאה, כנסים, קורסים מקוונים  
([Startup Eng.](#), [Master Prgrm.](#)), קהילות\פורומים,  
שפת תכנות חדשה, ...
- תרגול ([Kata](#), [Dojo](#), [Koans](#))
- קוד פתוח \ תרומה לקהילה
- הוראה
- משהו אחר בחיים... \ ניהול זמן



# Inspiring Jobs

*“Your work is going to fill a large part of your life, and the only way to be truly satisfied is to do what you believe is **great work**. And the only way to do great work is to **love what you do**. If you haven’t found it yet, keep looking. Don’t settle.”*



# (Self) Improvement Links

- Learning and self-improvement, Software Craftmanship [thread](#), 2010
- [The single most important thing you must do to improve your programming career](#), blog, 2008
- [10 טיפים שיהפכו אותך ממהנדס תוכנה למותג!](#), 2011
- [The Effective Workplace](#), 2011
- [www.kalzumeus.com/2011/10/28/dont-call-yourself-a-programmer](http://www.kalzumeus.com/2011/10/28/dont-call-yourself-a-programmer)
- [Why In-Person Socializing Is A Mandatory To-Do Item](#)

# מתי פרויקט בעצם נגמר?

• תשובות מבלוג [Artima](#):

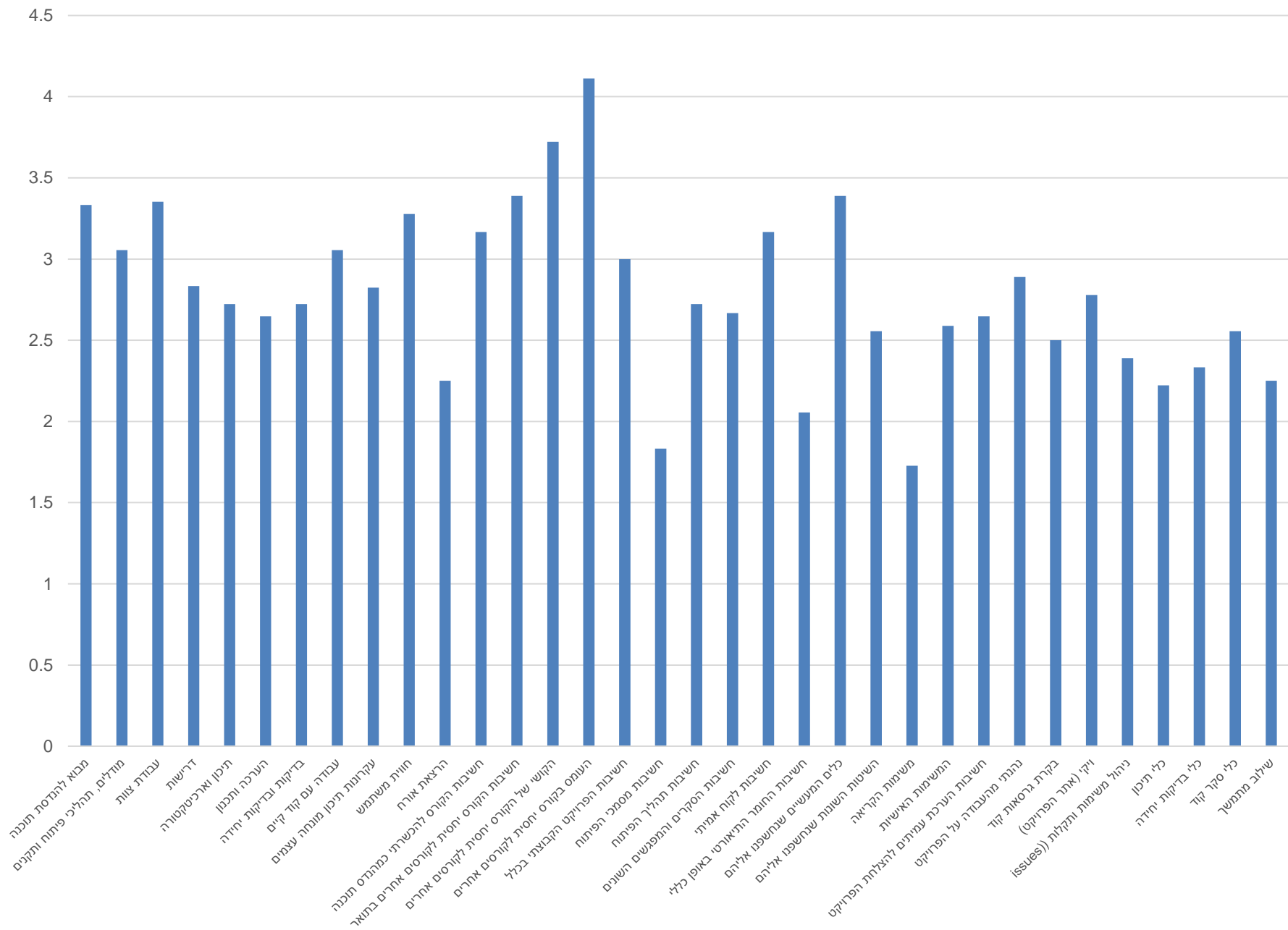
- When the deadline arrives or shortly thereafter
- "When I'm sick of working on it any longer"
- When the project is canceled and all the finger pointing stops
- [Stackoverflow](#): When the client/user signs off on the UAT (User Acceptance Test) after testing it ([SCRUM](#))
- [Hofstadter's Law](#): It always takes longer than you expect, even when you take into account Hofstadter's Law.

# עוד נושאים לסיכום

- פיתוח (כמעט) ללא קוד
- פיתוח (כמעט) ללא פרויקט
- פיתוח (כמעט) ללא עבודת צוות
- פיתוח (למרות) כישלונות \ מכשילים

# סקר

- 18 משיבים, 3 רצופים עוינים, חצי מהתשובות משתי קבוצות
- נושאים עיקריים בהערות:
  - git
  - עומס בקורס
  - הקדמת תכנות אינטרנט
  - הוספת חומר טכני בהרצאה
  - רלוונטיות\כלליות החומר התאורטי



# מבחן

## • מאגר שאלות (שיפורסם מראש)

– נושאים עיקריים מחומרי ההרצאה התאורטיים

– דיון והדגמה על הפרויקט שלכם

## • מקורות:

– חומרי ההרצאה (מצגות+מקורות+סיכומים)

– הפרויקט שלכם

## • פורום הקורס – פתוח לדיון

## • שיעור חזרה

# הצגת\שחרור גרסה סופית

