

"הזמן קצר והמלאכה מרובה..." – מסכת אבות
Eisenhower: "plans are useless, but
planning is everything"



Kent Beck @KentBeck · 22 שעות

The only way it's all going to go according
to plan is if you don't learn anything

הנדסת תוכנה

5. ניהול פרויקט: הערכה ותכנון

[Pragmatic Programmer Tip](#) :

Iterate the Schedule with the Code

Use experience you gain as you implement
to refine the project time scales.



השבוע

- תהליכים: ניהול פרויקט תוכנה
 - הערכה
 - תכנון
- שיטות וכלים
 - ~~Scrum~~
 - Github Issues: הזנת משימות, יצירת Backlog (project)
 - התחלת: בקרת גרסאות קוד – Version Control
 - ~~סקר SDS~~
 - ~~פרויקט שלב 5: סבב 0 – ZFR (רישום) – סקר~~
 - git init (לקראת ZFR ומשימה אישית 3)



הערכה - מקורות

- Cohn, Agile Estimating and Planning
- Agile Product Management with Scrum, chap. 3: [Working with the Product Backlog](#)
- Fundamentals of Software Project Estimation
<http://www.spc.ca/downloads/resources/estimate/fullestbasics.pdf>
- Pressman ch. 27
- [Evidence Based Scheduling](#) (Joel S.)
- The Art of Project Management: How to Make Things Happen
<http://msdn.microsoft.com/en-us/library/aa480154.aspx>
- COCOMO II <http://www.softstarsystems.com/overview.htm>
http://sunset.usc.edu/csse/research/COCOMOII/cocomo_main.html
- Pilone&Miles, Head First Software Development
- Kniberg, [Scrum and XP from the Trenches](#)
- Simula Research, <http://simula.no/news/simula-and-joergensen-top-ranked-within-software-engineering>

איפה אנחנו בפרויקט (בקורס)?

- למה?
בעיה (פלט: הצעת פרויקט\חזון\SOW)
- מי?
צוות (Inception, אתחול\תכנון פרויקט)
- מה?
דרישות (SRS)
- איך?
תיכון (ארכיטקטורה) (SDS)
- מתי?
ניהול, תכנון – (ZFR)
- בניה
(סבבי פיתוח - ~סקראם)



הלקוחות רוצים תוכנה (דרישות)
בזמן מסוים שבו הם צריכים
אותה
?





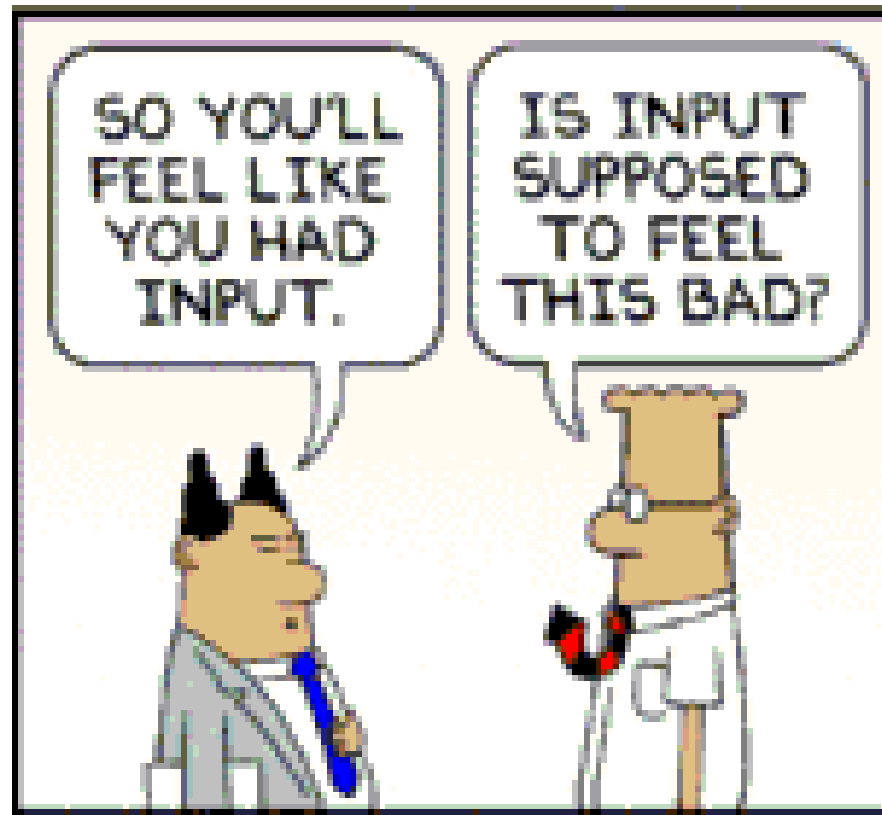












הערכה

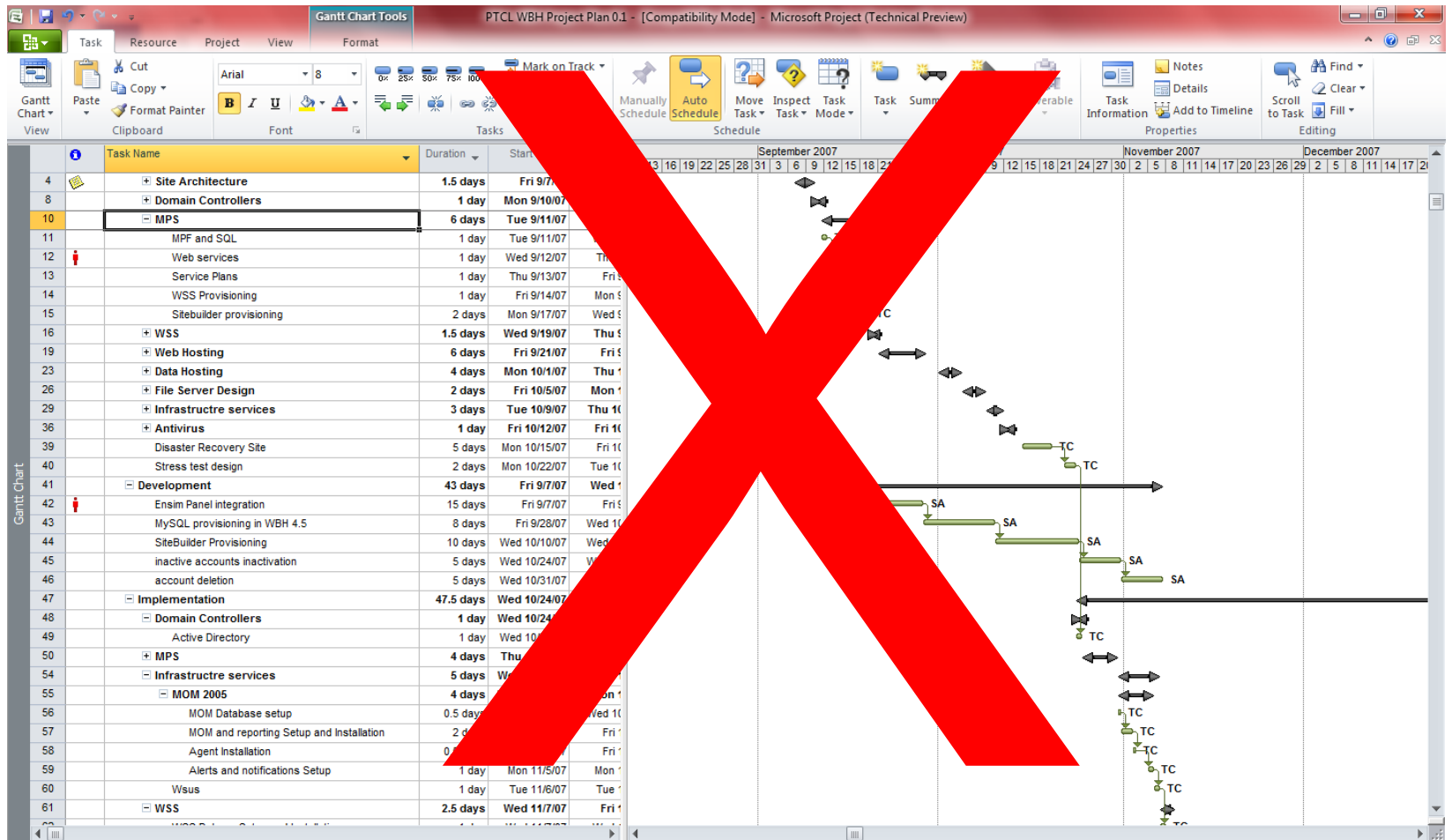
- מהי הערכה?
- למה זה חשוב? ולמה עכשיו?
- איך יוצרים הערכה?
- מה ההערכה שלכם?

מהי הערכה?

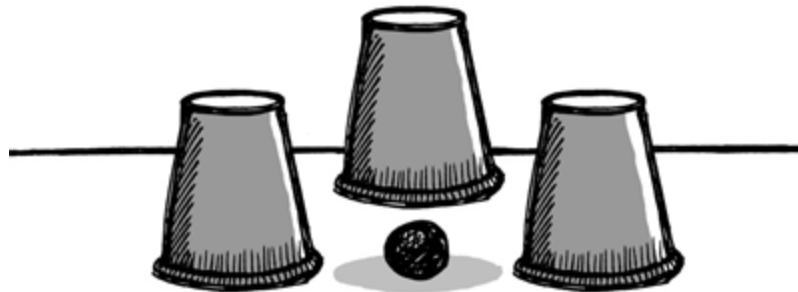
- פרדיקציה
- לא מטרה, לא התחייבות וגם לא...



וגם לא תכנון



**PLANNING
IS GUESSING**

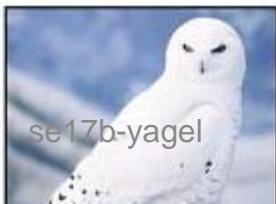




למה חשוב להעריך?

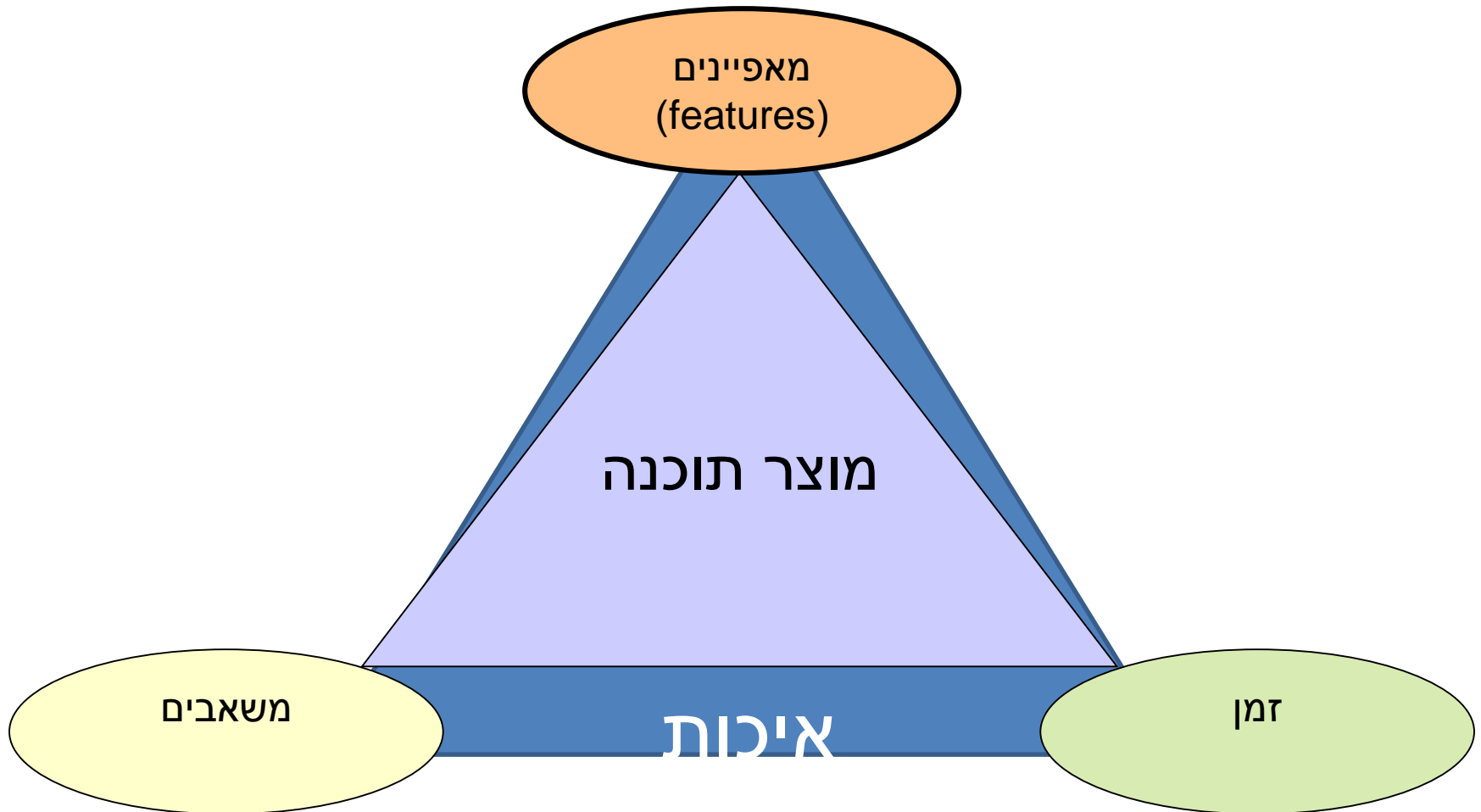
- לפי הניסיון, **מאד קשה** להעריך פרויקטי תוכנה
- מצד שני **תתבקשו** לתת הערכות ("זה יהיה מוכן כשאסיים...")
- הערכות **גרועות** יכולות להיות גרועות לכם, לצוות, לחברה, ללקוח
- Glass: "The two causes of runaways ... are **poor (usually optimistic) estimation** and unstable requirements" – p.28 (also van Genuchten '91)
- בואו נראה מה אפשר לעשות...

Facts and Fallacies of
Software Engineering



se17b-yagel

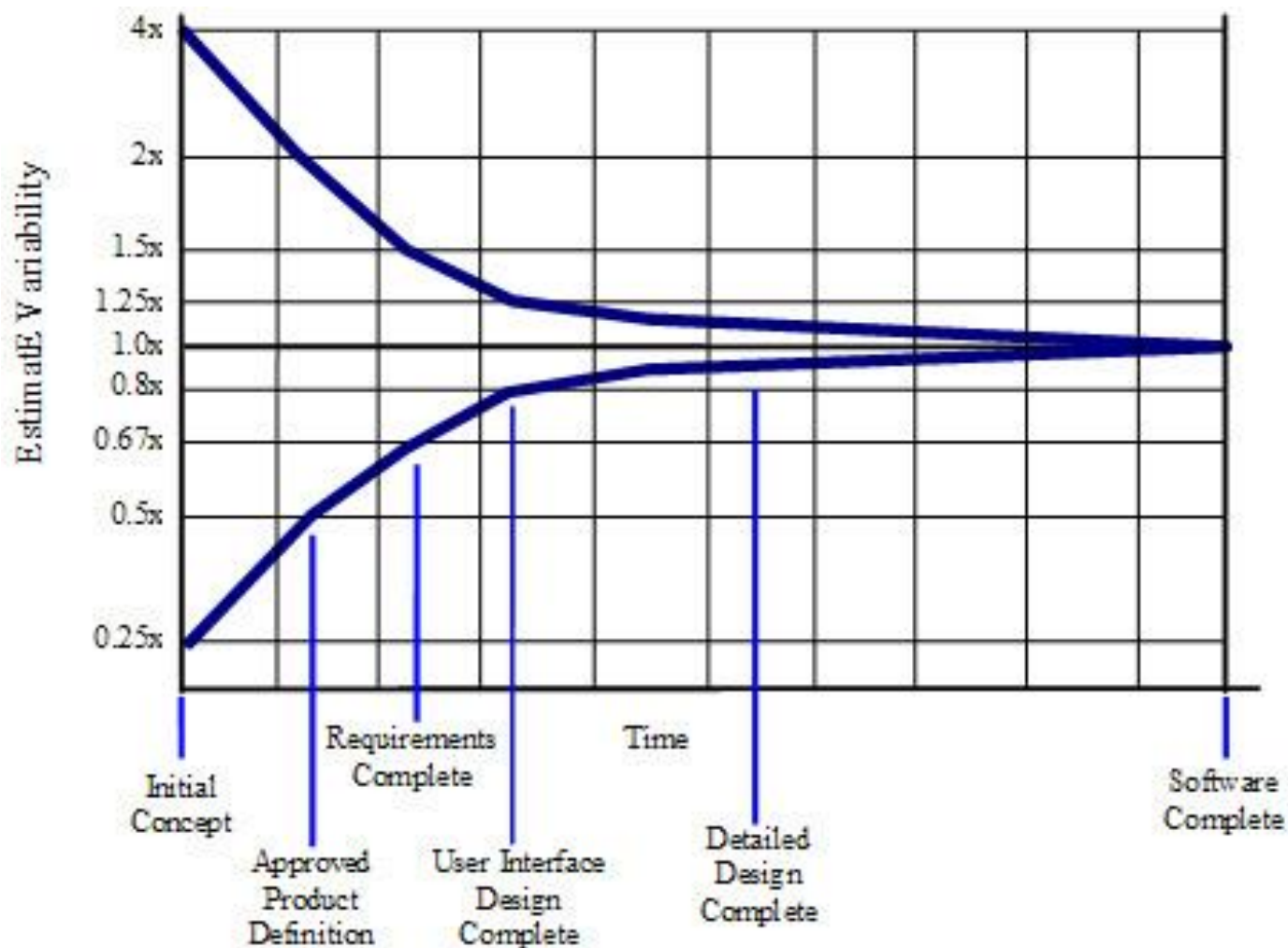
השפעת הערכה



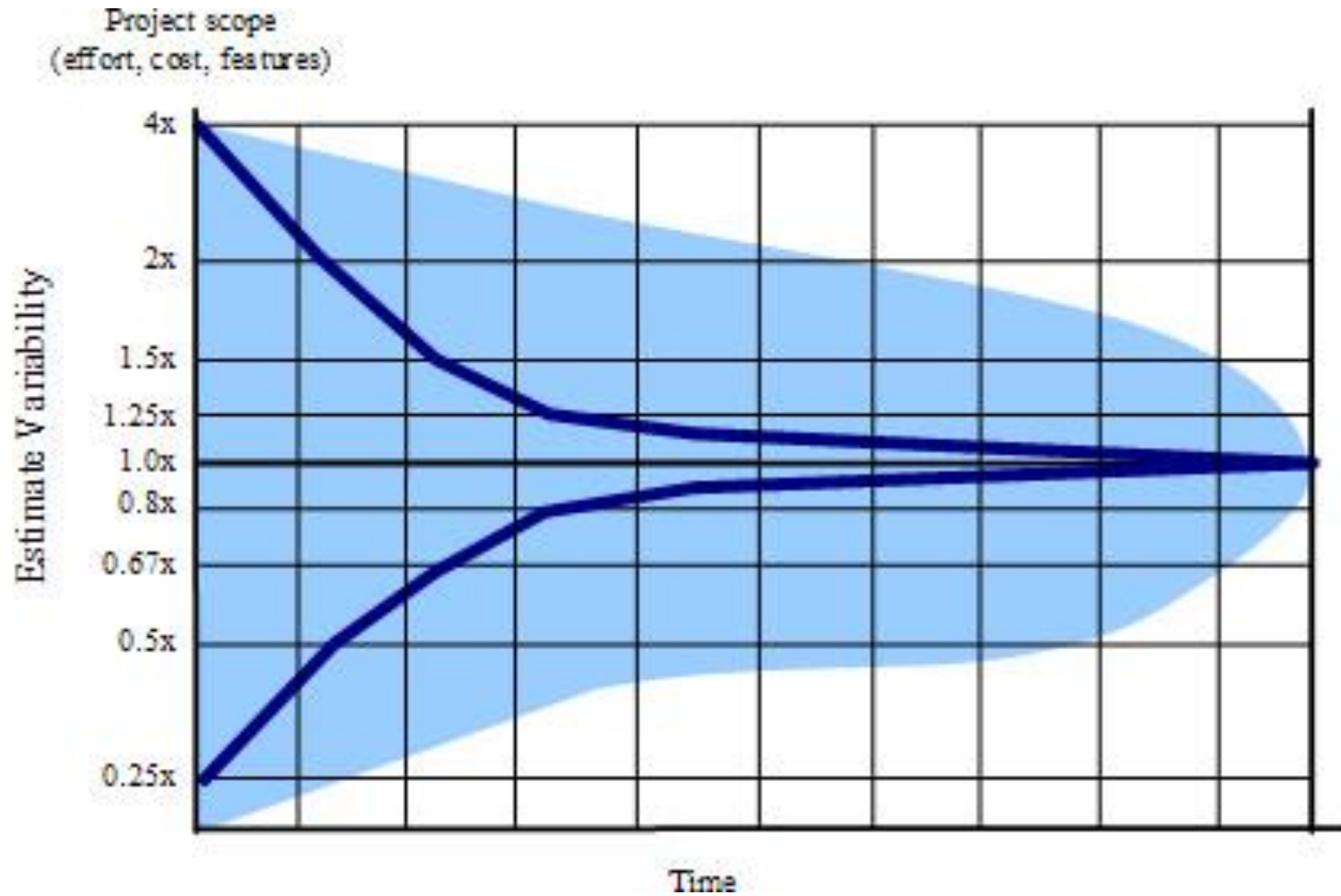
חוקי "יסוד"

- [חוק פרקינסון](#): משימה לוקחת את כל הזמן שניתן לה
- [סינדרום הסטודנט](#): כאשר ניתן זמן ארוך למשימה, מתחילים לעבוד עליה ברגע האחרון האפשרי... ומאחרים
- Planning Fallacy (D. Kahneman, [Wikipedia](#)):
“a tendency for people and organizations to **underestimate** how long they will need to complete a task, even when they have **experience** of similar tasks over-running.”

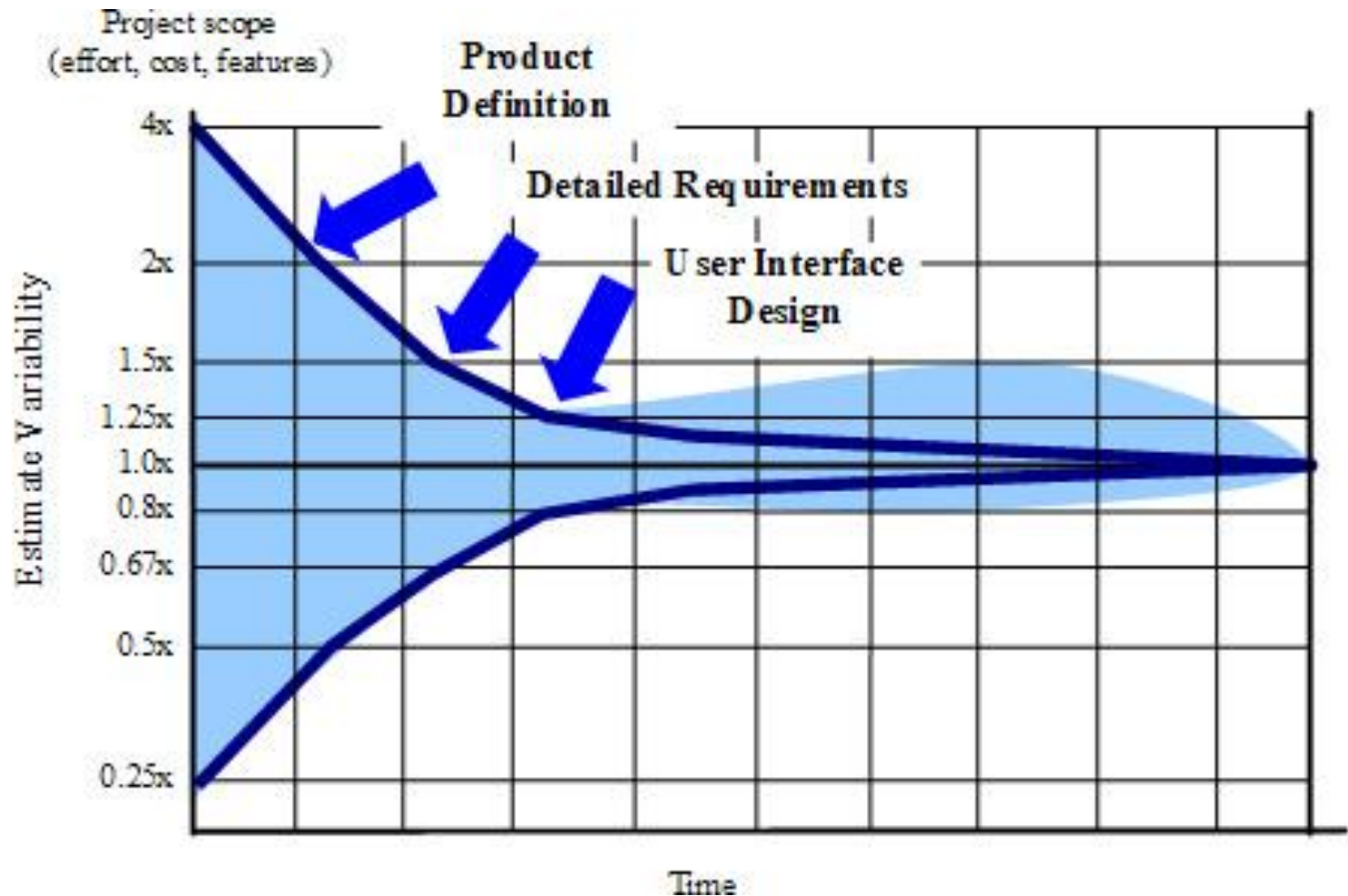
משפך אי-הוודאות [Boehm]



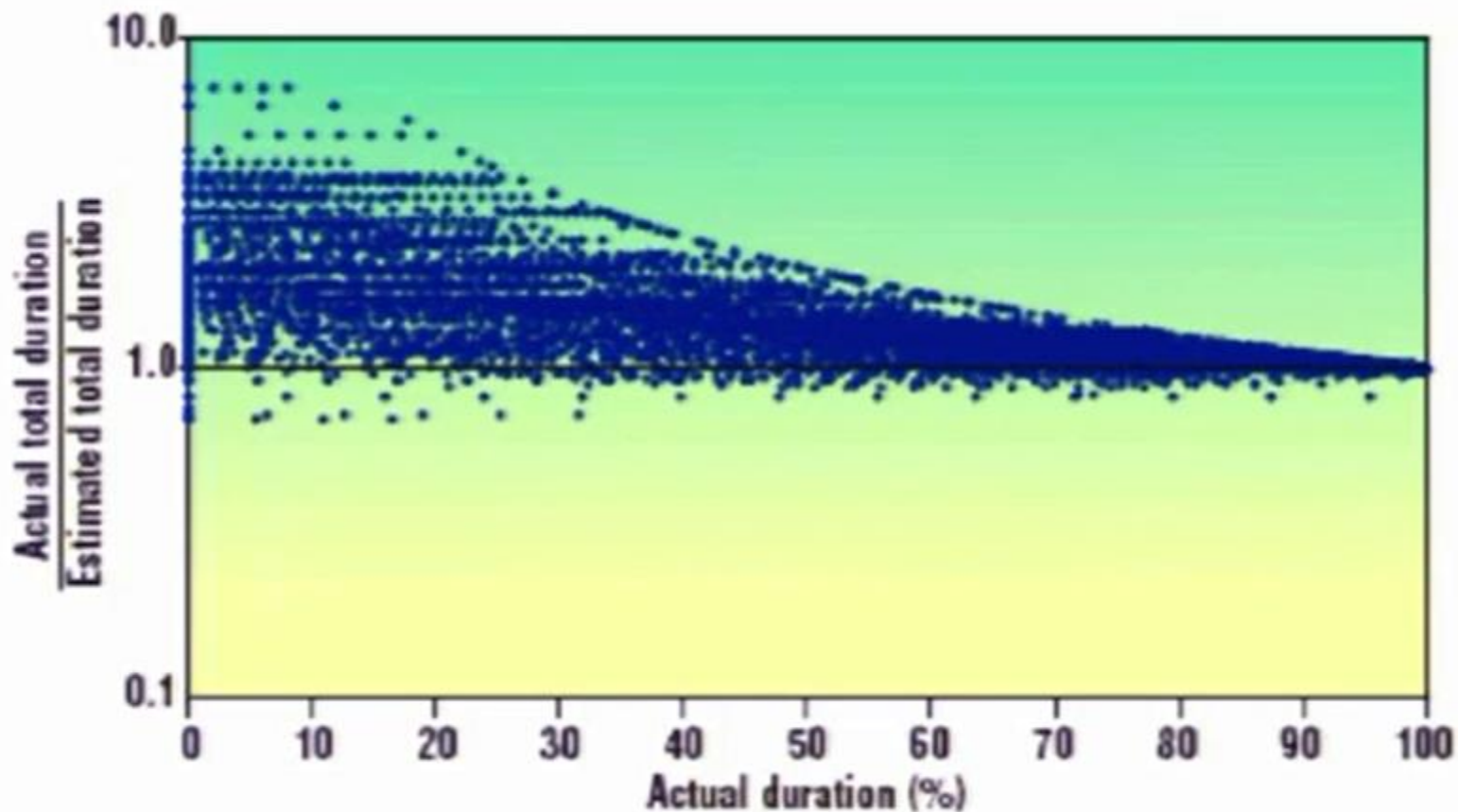
משפך אי-הוודאות



משפך אי-הוודאות

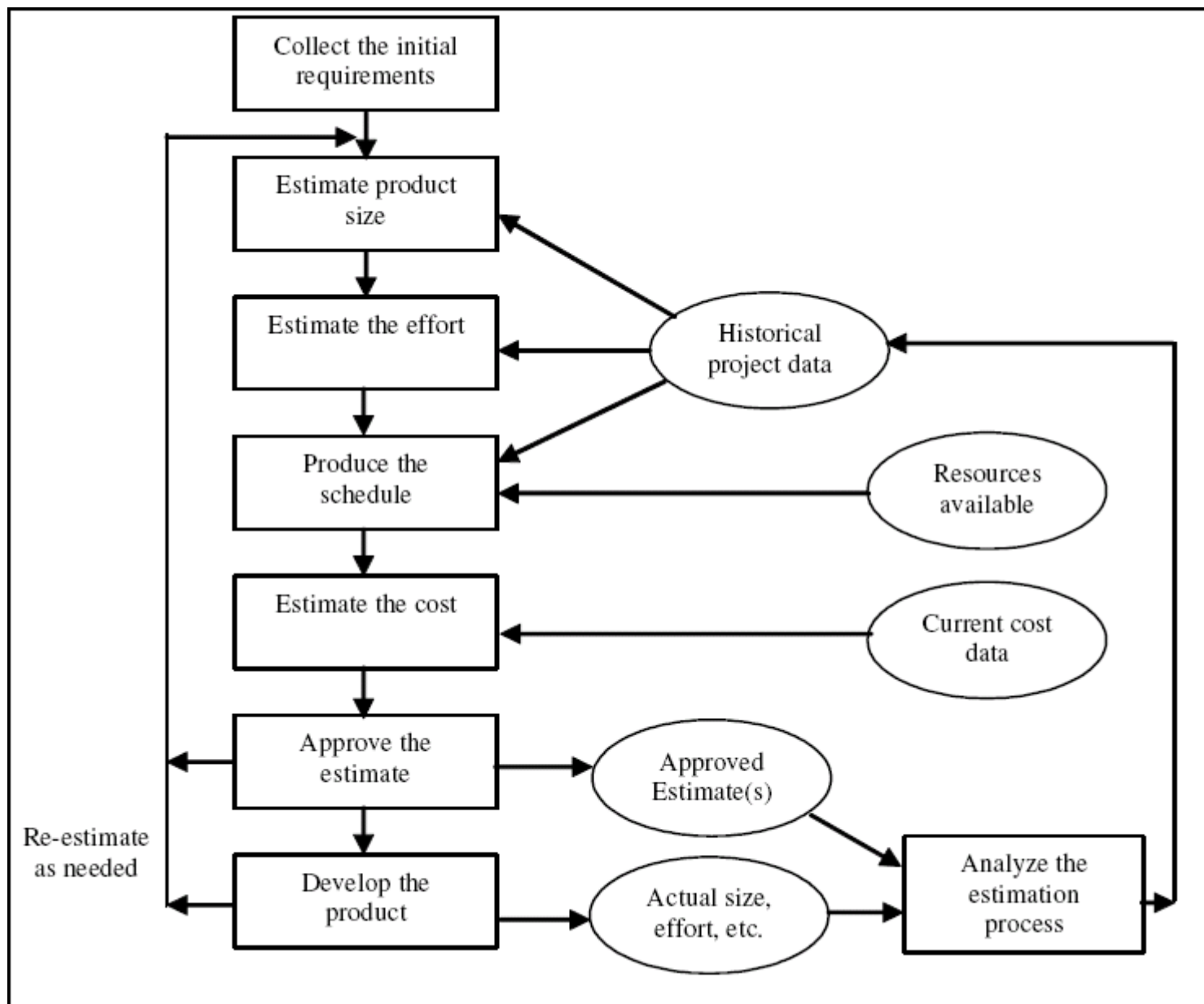


האם פרויקטים באמת מקדימים?



שלבי יצירת הערכה

1. הערכת **גודל** המוצר
 2. הערכת **המאמץ** הנדרש (חודשי\שעות אדם)
 3. הערכת **לוח הזמנים** בחודשי יומן
 4. הערכת **עלות** הפרויקט בש"ח
- עדכון ההערכות לפי התקדמות



1. הערכת גודל המוצר

- השוואה לפרוייקט בעבר
- הערכת מספר שורות קוד (SLOC)
- הערכת הפונקציונליות במוצר (Function Points)
- Use-case \ Story Points (RUP, Agile)
- התייעצות עם מומחים

האם SLOC מדד טוב?

- יש מתאם אך שונות גדולה
- תלוי בסוג הפרויקט (אינטרנט\זמן אמת)
- תלוי בשפת תוכנה*
- גודל (למשל שימוש חוזר)
- מפתח \ צוות

Language	Statements relative to C
C	1:1
C#	1:2.5
Fortran95	1:2
Java	1:2.5
Macro assembly	2:1
Perl	1:6
SQL	1:10

Source: Microsoft

2. הערכת המאמץ הנדרש

- תלוי בגורמים רבים ולא רק בגודל הפרויקט
- אפשר להשתמש בנתונים היסטוריים
- אם אין כאלו אז בגישה אלגוריתמית, למשל:
- COCOMO (Constructive Cost Model)
 - פותח ע"י Boehm בשנות ה-80
 - התפתח לגרסה 2 (agile +)
 - פשוט, כלים זמינים, מבוסס רגרסיה
 - הנחה: עלות הפרויקט תלויה בגודל הקבצים
 - לוקח בחשבון מאפיינים שונים



נוסחה בסיסית של COCOMO

- $\text{Effort (person/mon.)} = a \times K \text{SLOC}^b \times c$
 - a - organization factor
 - b - scaling factor / size penalty
 - c – product factor
- II התאמות נוספות:
 - דוגמאות להתאמת מאמץ (מתוך 17)
 - גודל DB
 - שמישות נדרשת
 - נסיון בשפה ובכלים
 - התאמת סדר הגודל (מתוך 5)
 - בשלות תהליכים
 - רוח הצוות

מחשבוני

- Basic:
 - http://sunset.usc.edu/research/COCOMOII/como81_pgm/cocomo81.html
- Real:
 - <http://csse.usc.edu/tools/COCOMOII.php>

שיטות נוספות להערכת מאמץ (+תזמון ומעקב)

- PERT
- [Wideband Delphi](#)
- Monte Carlo Simulation
- see [Wiki](#)pedia for many more

3. הערכת לוח הזמנים בחודשי יומן

- האם לא מספיק לחשב תאריך התחלה + מאמץ \ מספר מפתחים = תאריך סיום?
- Cocomo:
$$\text{Duration} = a * (\text{Effort})^b$$
 - Effort מהנוסחה הקודמת
 - מקדמי סדר גודל וצוות

הערכה אג'ילית

- מי שעושה את העבודה הוא זה שמעריך
- עבודת צוות
- שאיפה להסכמה (כיצד בד"כ יעריכו מתחילים?)
- “Less is more”
- הערכה בנקודות
- סדרות פשוטות (חזקות 2, פיבונאצ'י, T-shirt)
- משימה שחורגת שוברים לתת-משימות
- שיפור ההערכות תוך כדי - velocity

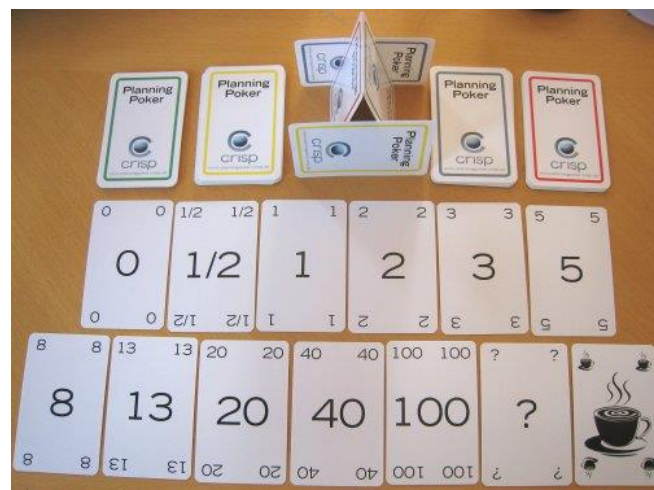
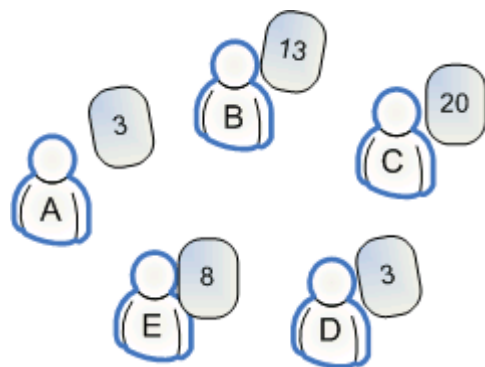
Agile Estimation

- הערכה לפני מימוש כל סיפור
- כל הצוות משתתף
- איך? למשל: פוקר התכנון

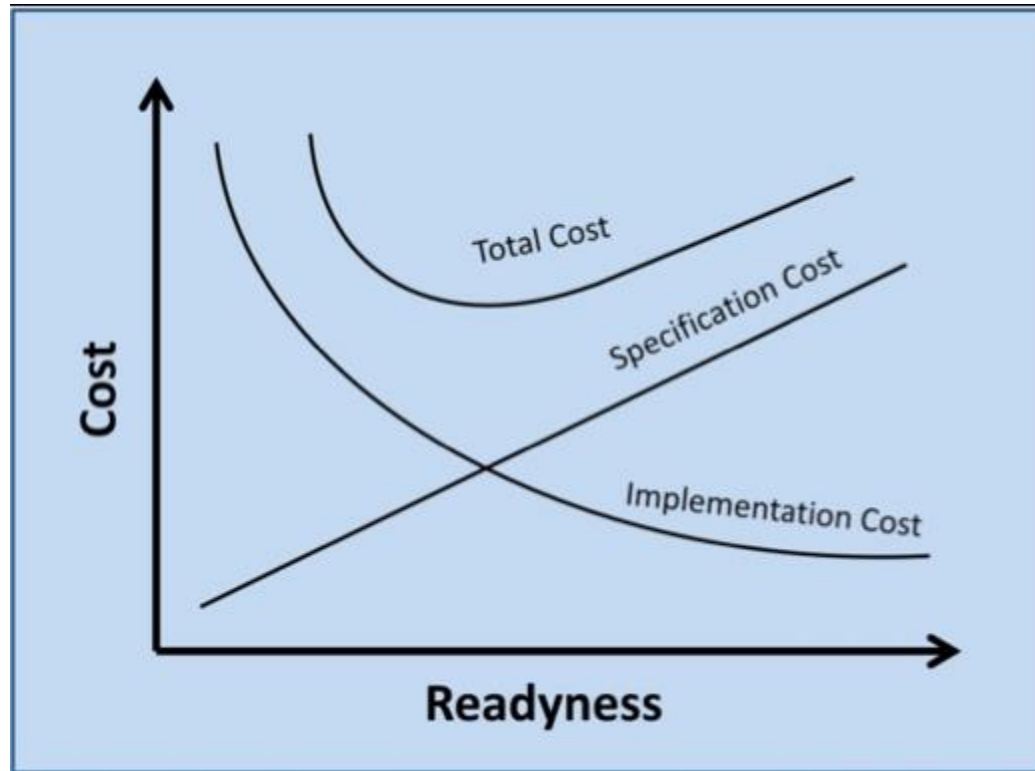
planningpoker.com

old.crisp.se/planningpoker

www.planitpoker.com/



כמה להשקיע בהערה ותכנון?



From: Boeg, [Real Life Scrum](#) (see also, Cohn 2010, [Pareto Principle](#) – 80/20)

From G. Wilson “Bits of Evidence”

Aranda & Easterbrook (2005): “Anchoring and Adjustment in Software Estimation”

*“How long do you think it will take to make a change to this program?”
(buried inside the spec)*

Control Group: *“I’d like to give an estimate for this project myself, but I admit I have no experience estimating. We’ll wait for your calculations for an estimate.”*

Group A: *“I admit I have no experience with software projects, but I guess this will take about 2 months to finish.”*

Group B: *“...I guess this will take about 20 months...”*

Results

Group A (lowball)	5.1 months
Control Group	7.8 months
Group B (highball)	15.4 months



The anchor mattered more than experience, how formal the estimation method was, or anything else.

Q: Are agile projects similarly afflicted, just on a shorter and more rapid cycle?

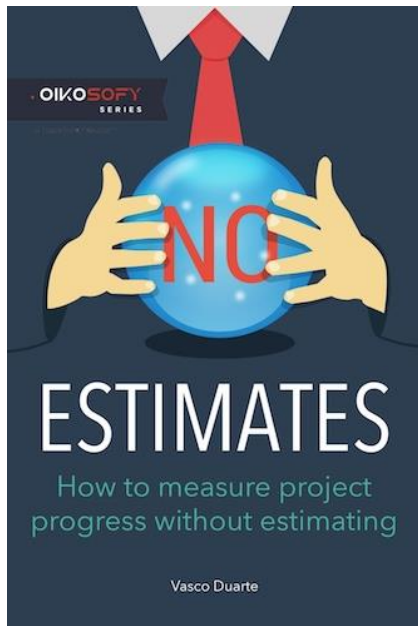
סיכום הערכה

- קשה להגיע להערכה טובה
- אך יש צורך למניעת הפתעות ותיאום עם אחרים
- לקחת בחשבון את סוג הפרויקט, הסביבה ועוד
- לנסות להימנע מטעויות...
- עוד נושאים

– הקצאת משאבים

– ניהול סיכונים

– #NoEstimates



Estimation Quotes

- “It’s better to be roughly right than precisely wrong.” – Keynes
- Brooks’ Law, “adding people to a late project only makes it later”
- What to say when asked for an estimate?
“I’ll get back to you” - Pragmatic Programmer, p. 68
- I will not ship SH%T
I will learn to say NO
When I give an estimate I will not LIE
- Robert Martin (SCNA’12):



2. תכנון \ Scrum (מדידה, בקרה ומשוב)





© 1994 United Feature Syndicate, Inc.

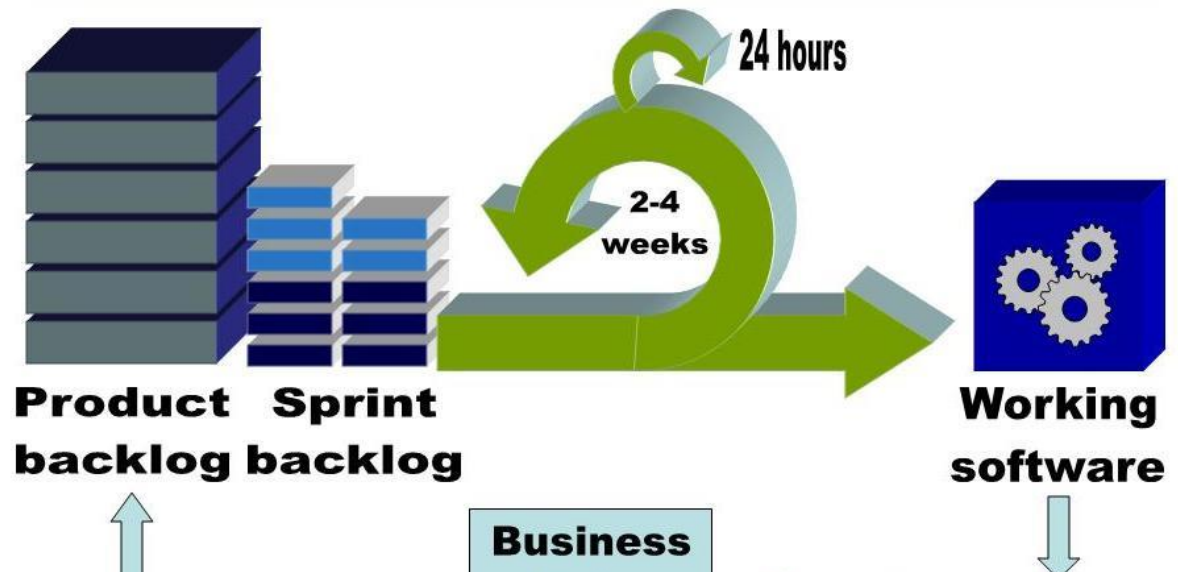


© 1994 United Feature Syndicate, Inc.



תהליך: SCRUM

- [Mike Cohn](#), [Getting Agile With Scrum](#)
- [מצגת בעברית \(*\)](#)
- [Scrum Guide](#): המדריך הרשמי:
- א. סופר, [סקראם בקיצור](#)

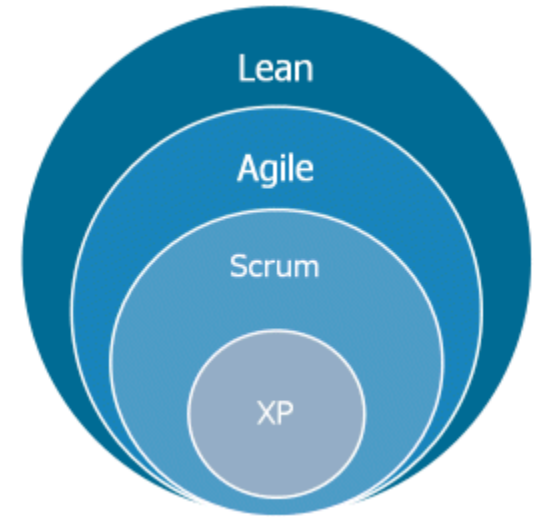


מה חסר בסקראם?

1. פרקטיקות פיתוח תוכנה ברמת הקוד
2. עמידות לשינויים תוך כדי פיתוח
3. התייחסות ללקוח של מוצר התוכנה
4. 1-3 נכונים

מעבר לסקראם

- Scrum Certification Debate
- Scaling Scrum
- Continuous Deployment (cont.)
- Lean/Kanban
- Pragmatics / craftsmanship
 - R. Martin, [The Land that Scrum Forgot](#)
 - M. Feathers, [The Mistake at the Heart of Agile](#)
 - M. Fowler, [It's Not Just Standing Up: Patterns for Daily Standup Meetings](#)
 - SAFe - [Scaled Agile Framework](#)
 - [The FrAgile Manifesto](#)
 - Video: [Spotify engineering culture](#)



SCRUM ואנחנו?

"[Steve](#) would draw a quick vision on the whiteboard, we'd go work on it for a while, bring it back, find out the ways in which it sucked, and we'd iterate, again and again and again. That's how it always went. **Iteration**. It's the key to design, really. Just keep improving it until you have to ship it.

- Some [critique](#) 😊

- מעבר לאחר ההתחלה
- Backlog / Sprintlog
- סבבים \ Sprints
- פגישות
- התפקידים השונים?
- Definition of Done?

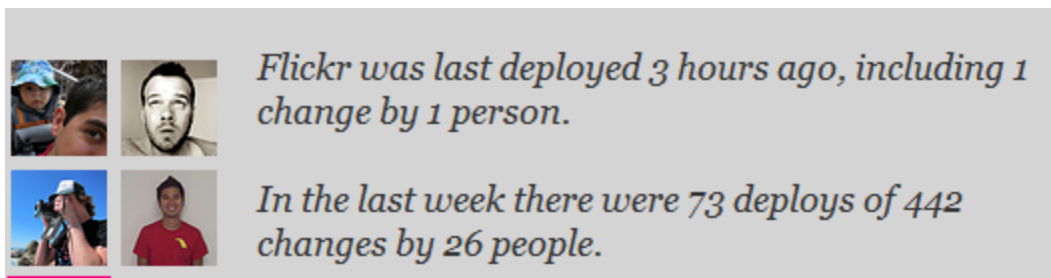
Kruchten: Colors in Your Backlog

Colours in a Product

	Visible	Invisible
Positive Value	Visible Feature	Hidden, architectural feature
Negative Value	Visible defect	Technical Debt

כיצד לחלק פרויקט?

- אבן דרך – milestone
- שחרור – release ([flickr 10/d](#), chrome, [facebook](#))
- איטרציה \ ספרינט
- יום \ שעה



- ראו גם:
http://en.wikipedia.org/wiki/Software_release_life_cycle
- אצלנו? סבבים

סבב 0 - ZFR

• Zero Feature Release

– רשימת נושאים \ סיפורים לעבודה – Backlog

• סבב 1 מפורט - [Sprint Backlog](#)

– מערכת בקרת גרסאות קוד (ומסמכים) – (תוך כדי)

• תשתיות כלליות של המוצר (אצלנו אב טיפוס)

– תחילת תיעוד, [Readme](#) ([Readme Driven Dev.](#))

• דיון

– מה מתאים לשים ב- [Backlog](#) – למשל [D.E.E.P](#)

– [האם בכלל צריך ZFR](#)? עוד [דיון](#)

MVP

- “The minimum viable product is that version of a new product which allows a team to collect the maximum amount of validated learning about customers with the least effort” (Reis / Lean Startup)
- [Minimum Viable Product](#) /
Minimum Marketable Feature / Main UC
[דוגמא](#)

סבב מס' 1 - MVP

- הצגת תרחיש עיקרי מוכן
- איך קובעים על מה לעבוד?
באחריות מי לתעדף?
- מה עושים אם לא מספיקים?
- משימות משנה לפי חומרי ההרצאה
- בדיקות יחידה, **Usability** / Refactoring, סקר קוד
Stable/Final Release
- בסיום כל סבב:
– רטרוספקטיבה (מה היה לנו?) ותכנון (מה הלאה?)

Walking Skeleton

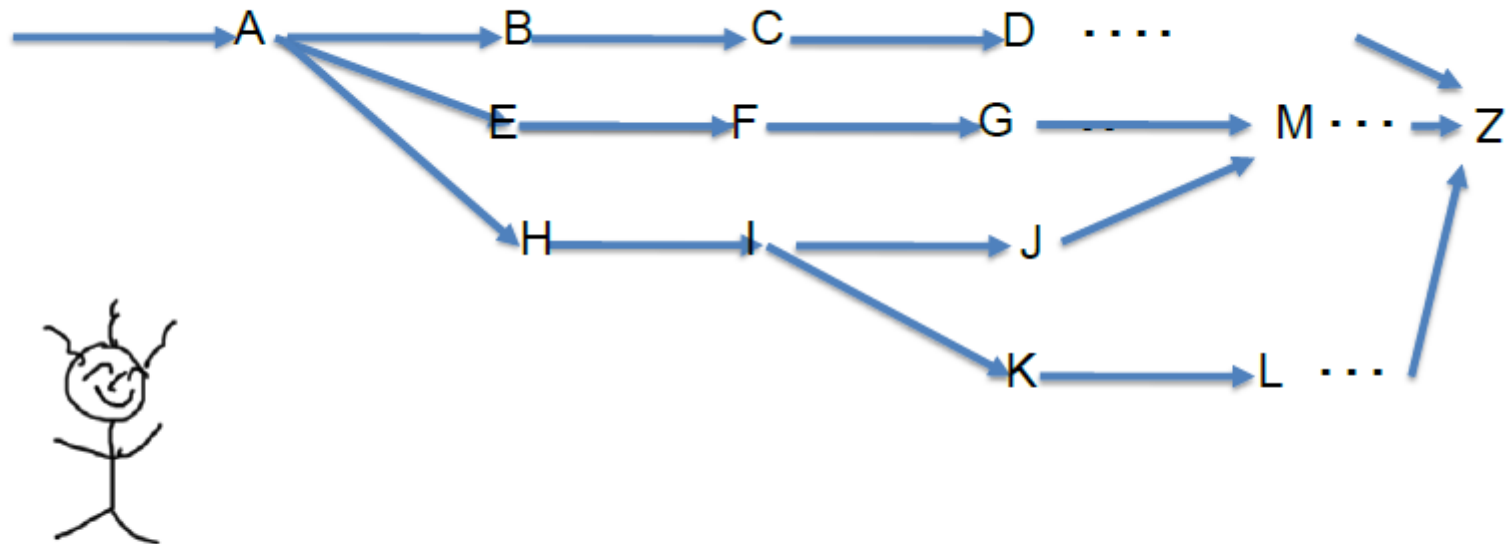
- שלד - ביסוס ארכיטקטורת המוצר:
 - התרחיש גם מתרגל את החלקים העיקריים
 - ב-production!
 - תקשורת ראשונית ומינימלית עם רכיבים חיצוניים
 - דגש על מקומות בעלי סיכון גבוה
- כבר לא מדובר רק באב-טיפוס
 - תיכון, בדיקות
- בכ"ז לא על חשבון המאפיין הראשון

סבבים נוספים

- אורך קבוע
- איזון בין אורך לכמות
- האם כל התפקידים בסקראם קיימים \ נצרכים?
- מה עושים כשלא מספיקים?
- TBC ...
- [Cockburn](#), “Simply using **iterations**, user stories and velocity doesn’t mean your project is **agile** – or on the way to **success**.”

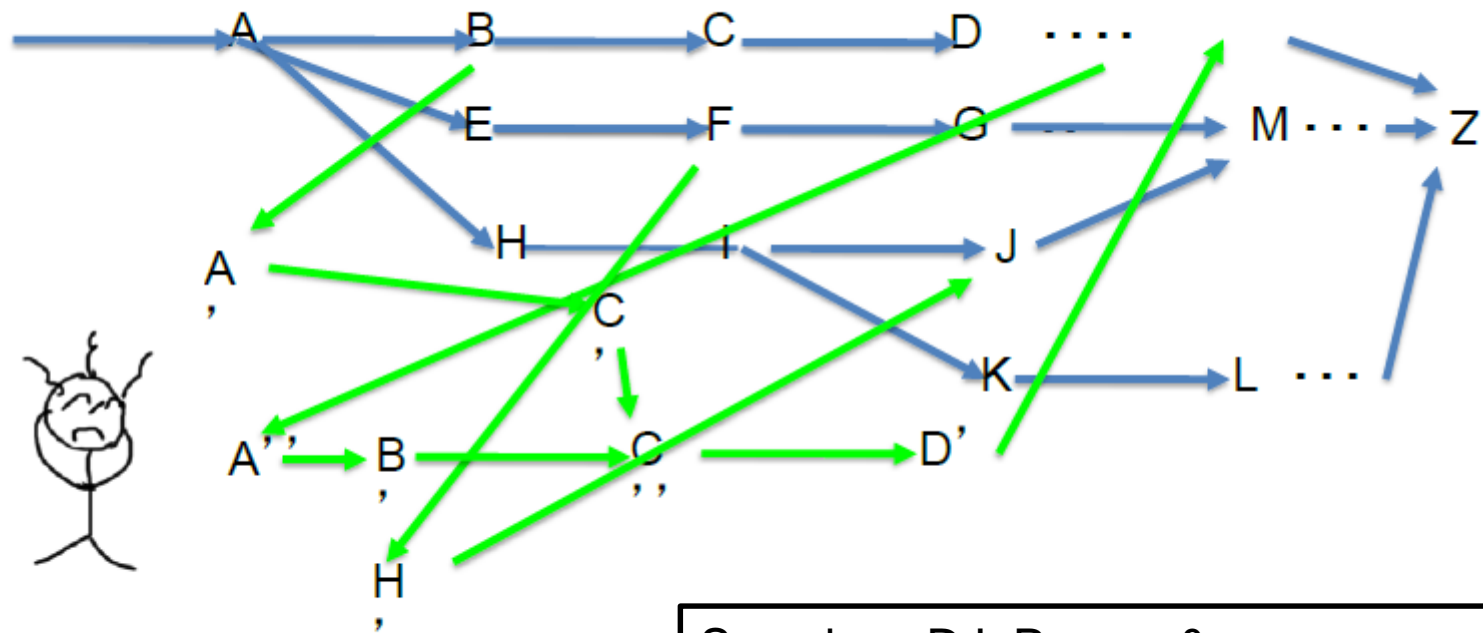
Plans vs. Reality (Rational vs. Irrational)

Plan



Plans vs. Reality (Rational vs. Irrational)

Reality



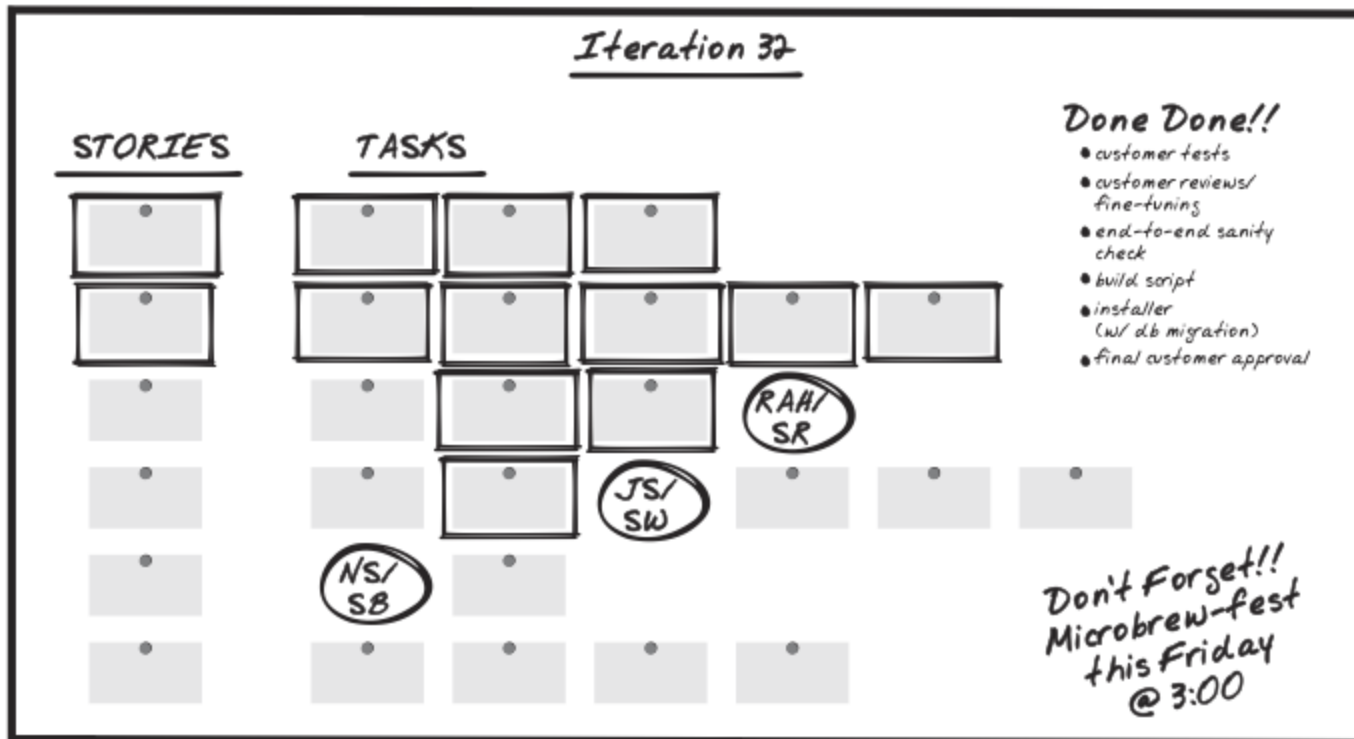
3. כלי עזר להנדסת תוכנה

- CASE: Computer Aided Software Engineering
- מערכת ניהול פרויקט ברשת, למשל GitHub
- אחרים: google-code ,CodePlex ,BitBucket
Zoho, ,<http://trac.edgewall.org/>
gforge.org ,redmine.org/
versionone.com ,pivotaltracker.com/
blossom.io ,visualstudio.com

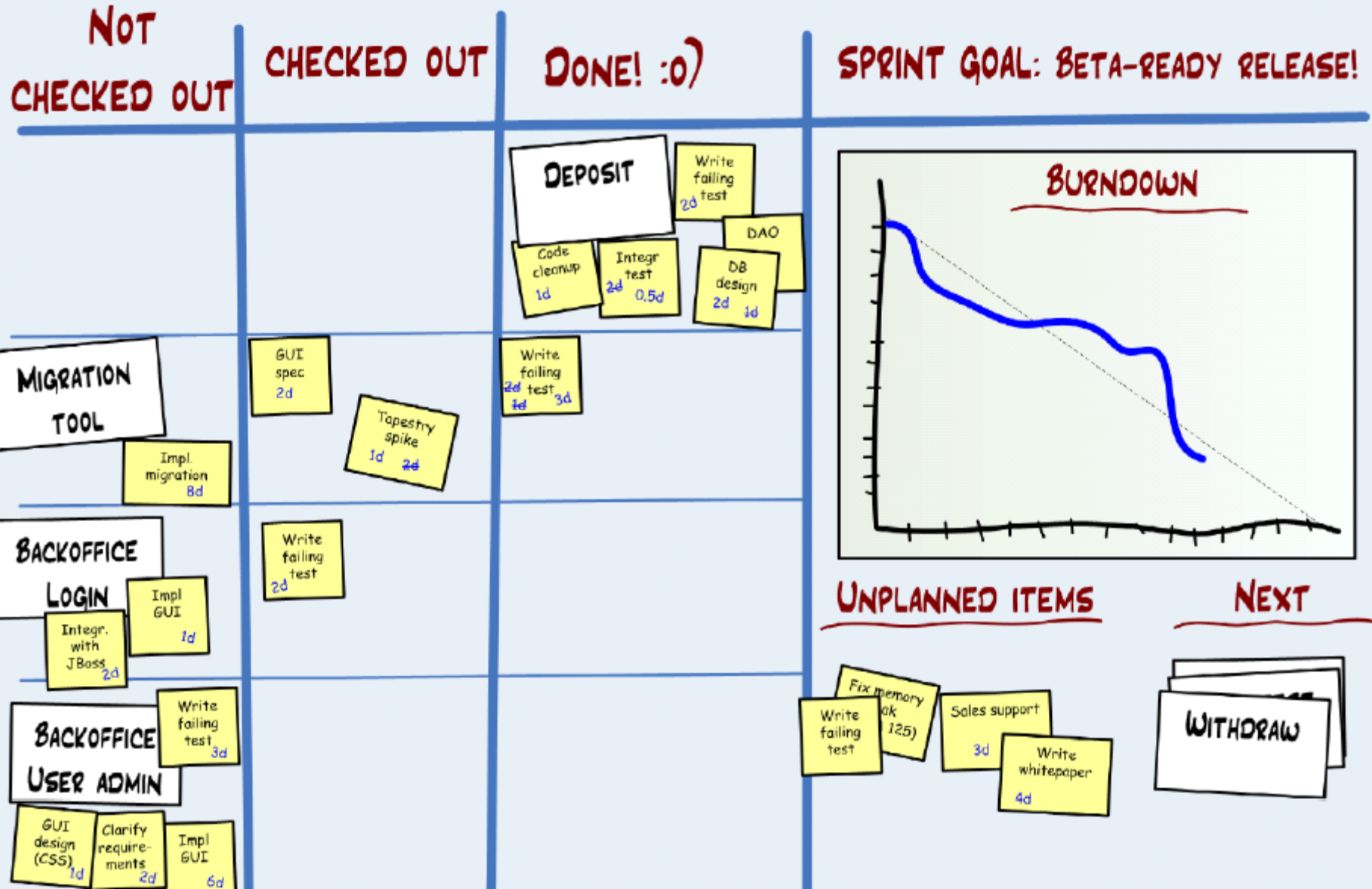
Planning with a Backlog

	Item #	Description	Est	By
Very High				
	1	Finish database versioning	16	KH
	2	Get rid of unneeded shared Java in database	8	KH
		- Add licensing	-	-
	3	Concurrent user licensing	16	TG
	4	Demo / Eval licensing	16	TG
		Analysis Manager		
	5	File formats we support are out of date	160	TG
	6	Round-trip Analyses	250	MC
High				
		- Enforce unique names	-	-
	7	In main application	24	KH
	8	In import	24	AM
		- Admin Program	-	-
	9	Delete users	4	JM
		- Analysis Manager	-	-
	10	When items are removed from an analysis, they should show up again in the pick list in lower 1/2 of the analysis tab	8	TG
		- Query	-	-
	11	Support for wildcards when searching	16	T&A
	12	Sorting of number attributes to handle negative numbers	16	T&A
	13	Horizontal scrolling	12	T&A
		- Population Genetics	-	-
	14	Frequency Manager	400	T&M
	15	Query Tool	400	T&M
	16	Additional Editors (which ones)	240	T&M
	17	Study Variable Manager	240	T&M
	18	Haplotypes	320	T&M
	19	Add icons for v1.1 or 2.0	-	-
		- Pedigree Manager	-	-
	20	Validate Derived kindred	4	KH
Medium				
		- Explorer	-	-
	21	Launch tab synchronization (only show queries/analyses for logged in users)	8	T&A
	22	Delete settings (?)	4	T&A

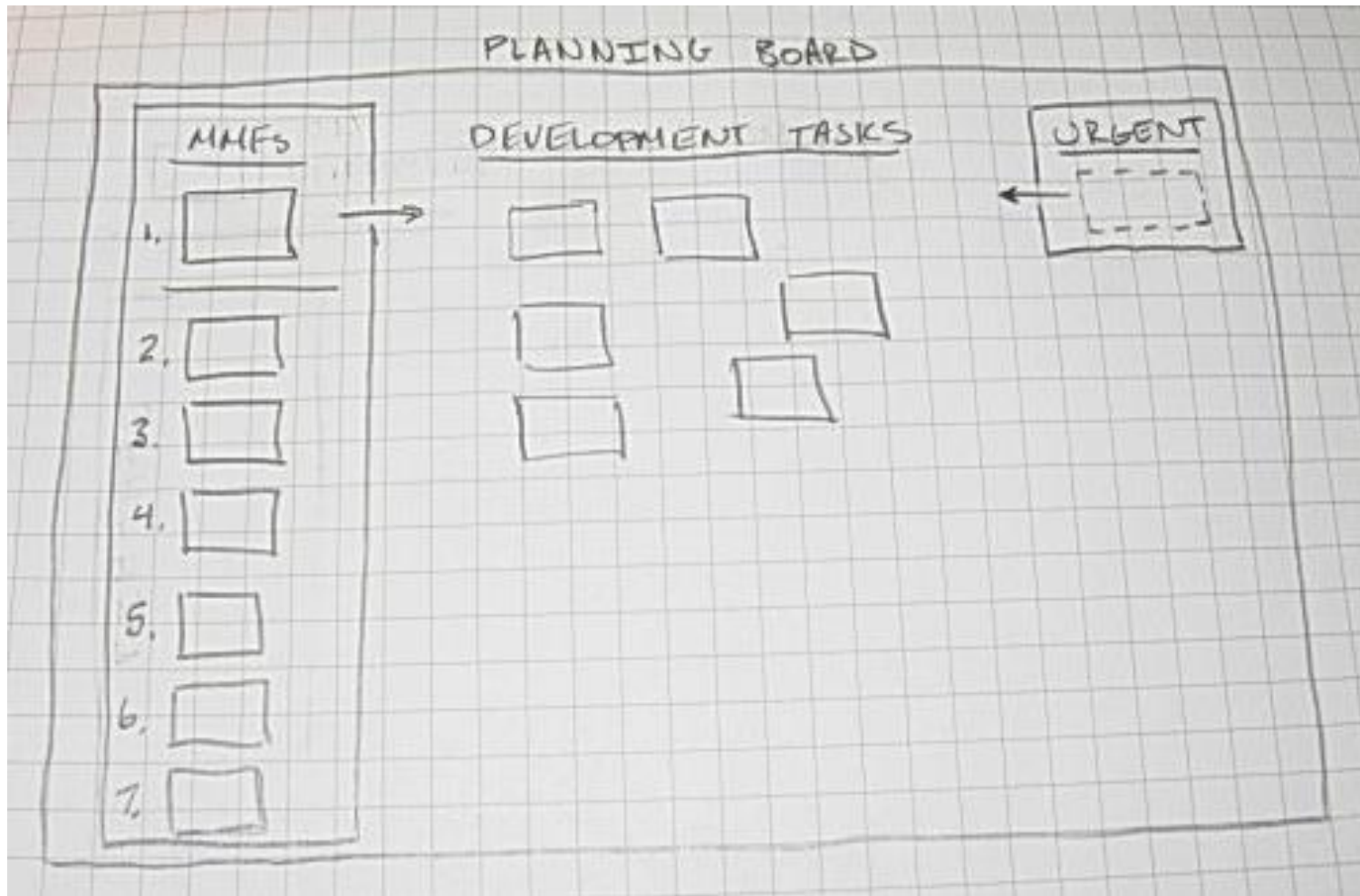
Iteration Planning Board



Burn Down Chart

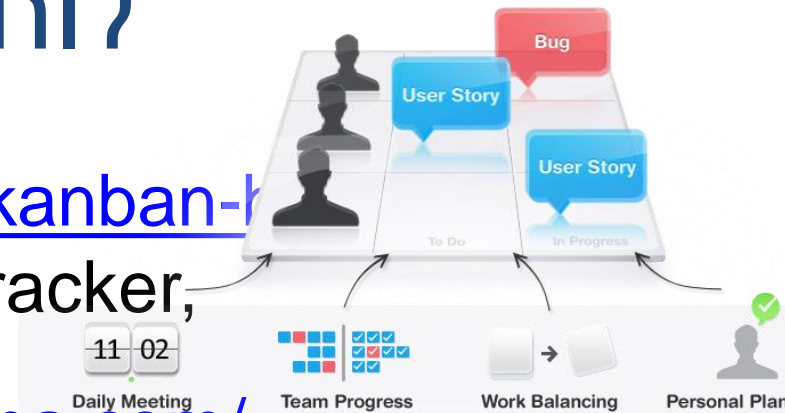


Kanban Board



לוחות מקוונים

- www.infoq.com/articles/agile-kanban-1
- Fogbugz, CodePlex, PivotalTracker, Team Foundation Service
<http://gissues.com/>, <http://asana.com/>,
leankitkanban.com, kickstarter.com,
agilezen.com/, jetbrains.com/youtrack/,
leankitkanban.com remindmethis.com,
cardboardit.com, [mingle](http://mingle.com), overv.io
- Simple g-spreadsheet board kanban.ws/
trello.com (used e.g. by uservice, no sprints)



• בפרויקט: דף סבב ויקי + github issues + לוח Huboard

Github

- ויקי
- ~~הורדות~~, שחרורים
- חברתי \ גרפים, תרומות
- ✓ **ניהול משימות**
- **בקרת גרסאות**
- סקרי קוד (בהמשך)
- קישור לאתרי **רזומה**
- חינם לקוד פתוח

CAREERS 2.0
by stackoverflow



+



Have projects on GitHub?
Import them easily to your profile

פרויקט 5 - ZFR

Git init

- <http://stackoverflow.com/research/developer-survey-2015#tech-sourcecontrol>
- – בוטל מ - 2016
- try.github.com
- gitimmersion.com
- Git (GUI) Clients
- הרצאה מומלצת:
https://github.com/jimweirich/presentation_source_control

בפעם הבאה

- בקרת גרסאות + git (להביא מחשב)
- פרויקט: סבב 0 - ZFR
 - תכנון המשך הפרויקט
 - העלאת תשתית קוד ראשונה
 - תכנון סבב 1
- סקר ZFR בשבוע הבא (הרצאה / תרגיל, יש להירשם)
- המשך: סבבי פיתוח
- משימה אישית 3 (בשלבים)

לסיכום

- תהליכים: הערכה ותכנון
- שיטות: Cocomo, פוקר תכנון
- כלים – לוח קנבן ועוד (האם תורמים? האם יתרמו?)
 - Github, Git
- נושאים נוספים:
 - מדדי תוכנה (ולמה הם מסוכנים)
- הרצאות המשך:
 - בקרת קוד
 - בדיקות, חווית משתמש
 - עקרונות עיצוב מונחה עצמים
 - כלים ושיטות נוספים...