

פרק 3

הגדרה וניהול של דרישות

Requirements Definition and Management



פעילות הגדרת דרישות הלקוח ובעלי העניין

• מטרת הפעילות

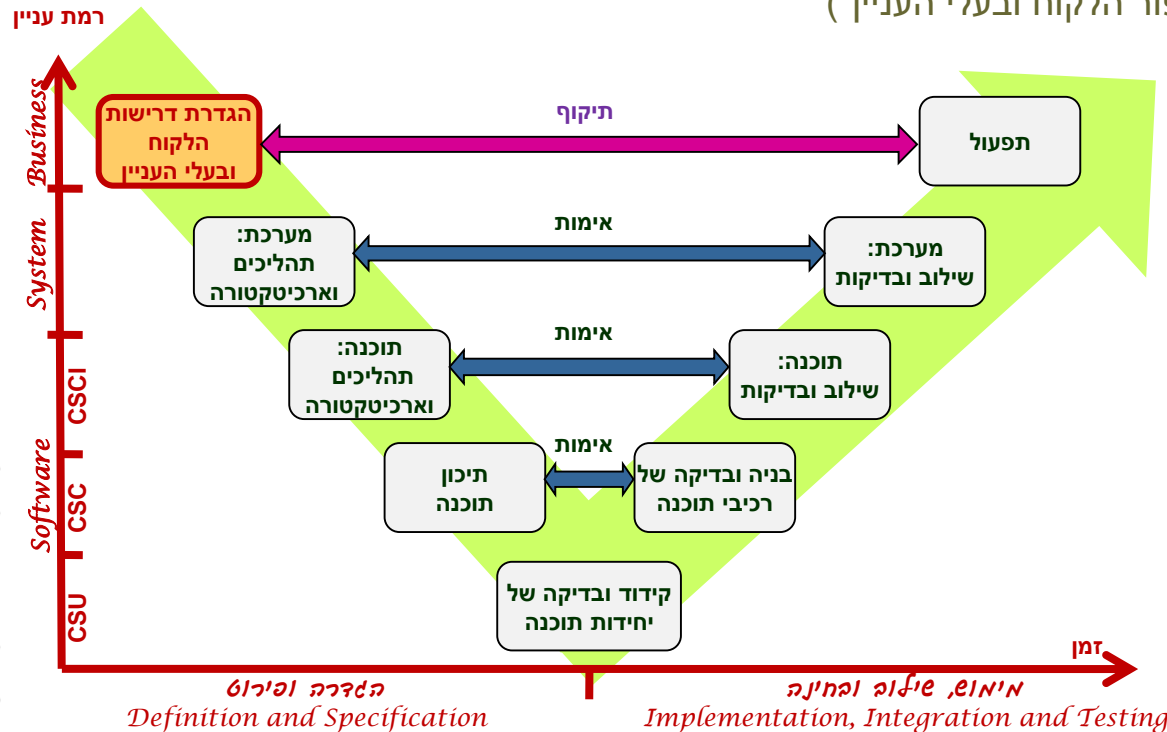
- הבנת הבעיה / הצורך + תפיסת הפתרון
- גיבוש אוסף דרישות, על פי סוגים
- משקף את מרכיבי הבעיה / הצורך ותפיסת הפתרון
- מהווה בסיס (קו ייחוס) לניהול תהליך הפיתוח

• קלט

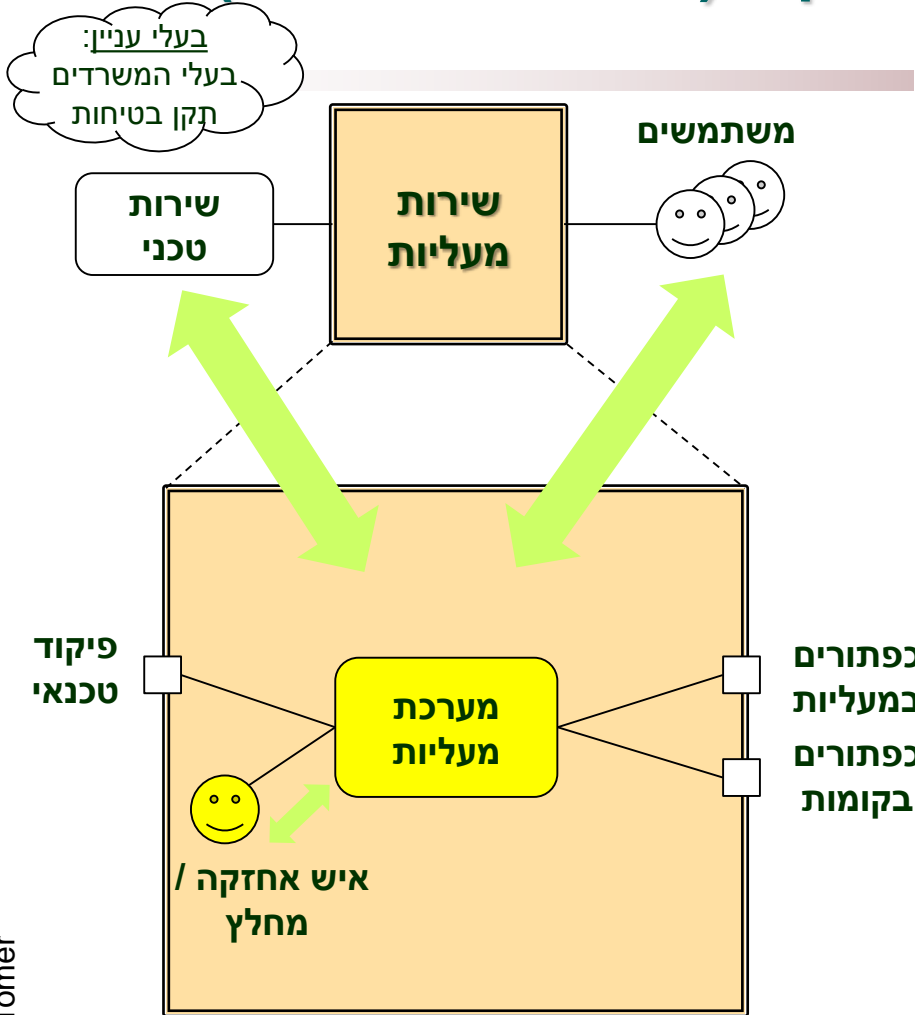
- אפיון טכני, אפיון תפעולי ("סיפור הלקוח ובעלי העניין")

• תוצר

- טבלת דרישות, על פי סוגים



שירות מעליות – הרמה הארגונית/עסקית (Business Level)



• מטרה (מה ה-business של שירות המעליות?)

- הסעת אנשים בין קומות
- + חילוץ במקרה הצורך (שירות חילוץ פנימי)
- + תחזוקה שוטפת (שירות טכני חיצוני)

• סביבה

- משתמשים
- שירות טכני

• מרכיבים

- מערכת מעליות
- איש אחזקה / מחלץ

• ארגון

- ממשקים חיצוניים למשתמשים
- כפתורים במעליות ובקומות
- ממשק לפיקוד טכנאי
- ממשק פנימי לאיש האחזקה / מחלץ

• אינטראקציה

- חיצונית
- עם משתמשים חיצוניים לצורך מילוי בקשות נסיעה
- עם שירות טכני לצורך בדיקה ותיקון
- פנימית
- עם איש אחזקה לצורך חילוץ

שירותי המעליות – אפיון טכני

• שירותי המעליות בבניין המשרדים כולל את האלמנטים הבאים:



- 3 מעליות המשרתות 10 קומות
 - לכל מעלית יש בקר (מחשב) עצמאי + תוכנת בקרה למעלית
 - בכל מעלית יש פאנל לנוסע ובו כפתורים כדלהלן
 - כפתורים לבקשת נסיעה לקומות 0 1 2 ... 9
 - כפתור לעצירת חירום
 - כפתור לאזעקת חילוץ
 - בכל קומה יש כפתור אחד לעליה וקפתור אחד לירידה
 - פרט לקומה העליונה (לירידה בלבד) ולקומת הקרקע (לעליה בלבד)
 - חדר מכונות בתוכו נמצאים
 - מחשב + תוכנה לפיקוד ובקרה על כל המעליות
 - פאנל הפעלה למחלץ לצורך ביצוע פעולת חילוץ למערכת שנתקעה
 - המחלץ הוא איש אחזקה של הבניין, ומהווה חלק מהשירות
 - פאנל הפעלה לטכנאי לצורך בדיקה ותיקון של המערכת
 - הטכנאי איננו שייך לבניין אלא לשירות טכני חיצוני

שירות מעליות – אפיון תפעולי (סיפור הלקוח ובעלי העניין)

נוסע הנמצא בקומה כלשהי ורוצה להזמין מעלית לוחץ על הכפתור המתאים לכיוון הנסיעה המבוקש. אם לא היה דלוק קודם לכן, הכפתור נדלק בעקבות הלחיצה. מעלית כלשהי הנמצאת בכיוון הנסיעה המבוקש תגיע לקומה, תוך דקה לכל היותר. עם הגעתה נפתחת הדלת והכפתור כבה.

נוסע הנמצא בתוך המעלית ורוצה להגיע לקומה כלשהי לוחץ על הכפתור המתאים לקומה. אם לא היה דלוק קודם הכפתור נדלק בעקבות הלחיצה ולמעלית נוספת בקשת עצירה. הדלת נסגרת, לאחר השהיה, והמעלית ממשיכה בנסיעה, כאשר היא עוצרת בכל קומה עבורה קיימת בקשת עצירה. כאשר המעלית נעצרת בקומה הדלת נפתחת והכפתור המתאים לקומה כבה.

נוסע במעלית יכול לעצור את המעלית בזמן נסיעה באמצעות לחיצה על כפתור עצירת החירום. במקרה זה המעלית עוצרת מיד וכל בקשות העצירה שלה מתבטלות. לאחר מכן ניתן להחזיר את המעלית לפעולה על ידי לחיצה על כפתור עבור קומה כלשהי.

במקרה שהמעלית נתקעה במהלך נסיעה מזעיק הנוסע חילוץ באמצעות כפתור החילוץ. המחלץ (שהוא איש האחזקה של הבניין) מגיע לפאנל החילוץ שבחדר המכונות ומפעיל פקודות להורדת המעלית לקומת הקרקע ולפתיחת הדלת.

איש האחזקה אחראי להעלות את המערכת (Start Up) בתחילת יום פעילות ולהוריד אותה (Shut-Down) בסיומו.

טכנאי, המגיע אחת ל-6 חודשים, יכול לבצע בדיקה מקיפה של כל המערכת ולתקן תקלות באמצעות פיקוד הטכנאי שבחדר המכונות.

מערכת המעליות תעמוד בכל תקני הבטיחות הישימים.

המערכת תונגש לבעלי מוגבלויות שונות.

בעלי עניין – ה"דורשים" של הדרישות

- בעל עניין (Stake Holder) - הגדרה

- גורם המושפע מפיתוח המערכת או אחראי באופן כלשהו על פיתוחה

- בעלי עניין אופייניים למערכות עתירות תוכנה

- משתמשים (Users)

- מי שיפעילו את המערכת לצרכיהם (סוגים שונים עם צרכים שונים)

- לקוחות (Customers, Clients)

- מי שהזמינו את המערכת וישלמו עבורה

- מנתחי שוק (Market Analysts)

- מי שמייצגים את הלקוחות והמשתמשים של השוק ומנתחים את צרכיהם ודרישותיהם

- רשויות תחקיקתיות (Regulators) וגופים מקצועיים

- מי שמגדירים תקנים או חוקים הישימים לפיתוח התוכנה

- מהנדסים (Engineers)

- מי שמייצגים אינטרסים טכניים המשפיעים על דרישות המערכת, כגון טכנולוגיות ישימות, שימוש חוזר, שימוש ברכיבים-מן-המדף וכדומה

- מנהלים (Managers)

- מי שמייצגים אינטרסים של הארגון או הפרויקט המשפיעים על דרישות המערכת, כגון שיתוף משאבי-אנוש, תיאום עם פרויקטים אחרים, יעדים עסקיים של הארגון, לו"ז ותקציב

לבעלי העניין יש זוויות ראייה שונות על המערכת...



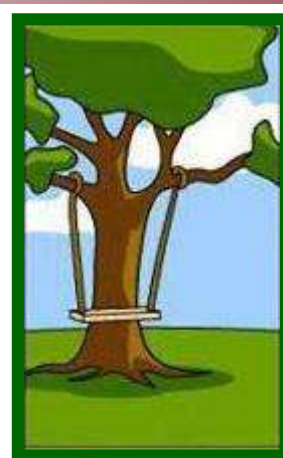
מה שתארו
אנשי השיווק



מה שקודדו
התכנתים



מה שתכנן
מנתח המערכת



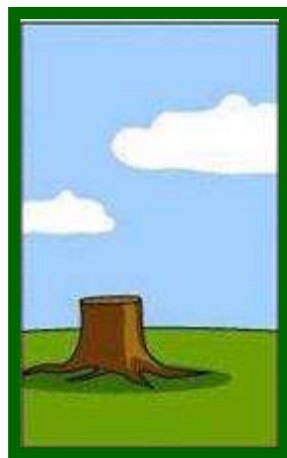
מה שהבין
מנהל הפרויקט



מה שביקש
הלקוח



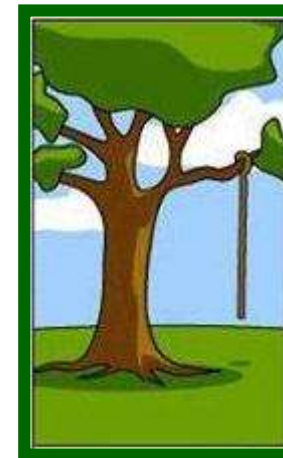
מה שהלקוח באמת
היה צריך ...



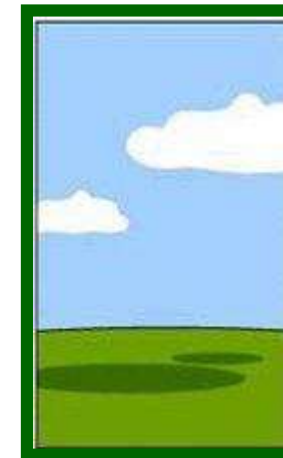
מה שכיסתה
התמיכה



מה שנכלל
בחשבון ללקוח



מה שהותקן
אצל הלקוח



מה שתועד

מהיכן באות הדרישות? (1)

- ראיונות/שיחות עם בעלי עניין

- חשיפת מידע מבעלי עניין באמצעות שיחה ישירה איתם



- ראיון רשמי / לא-רשמי

- שאלות מוכנות מראש / שאלות ספונטניות

- אחד-על-אחד / ראיונות קבוצתיים

- סדנאות מונחות (Facilitated Workshops)

- Quality Function Deployment (QFD)

- הגדרת המאפיינים הקריטיים (תפקודי האיכות) של הפרויקט על בסיס "קול הלקוח"

- (Voice of the Customer) ואילוצים ומגבלות של המפתח

- בקשות להצעות (RFP), מכרזים והצעות

- תפיסת המערכת הרצויה בעיני הלקוח

מהיכן באות הדרישות? (2)

- **תקנים ורגולציות**

- תקנים כלליים לדיסציפלינות (למשל תקן לפיתוח תוכנה ISO 12207)
- רגולציות ספציפיות לתחום (למשל FDA, FAA)
- נהלים ארגוניים

- **מפרטי תכן קודמים**

- שלבי ניתוח ותכן מוקדמים (תכן מערכת-על)
- מפרטים של מערכות דומות שפותחו בעבר

- **אבות-טיפוס (prototypes, demos)**

- מודל "עובד" של המוצר הצפוי טרם בנייתו המעשית

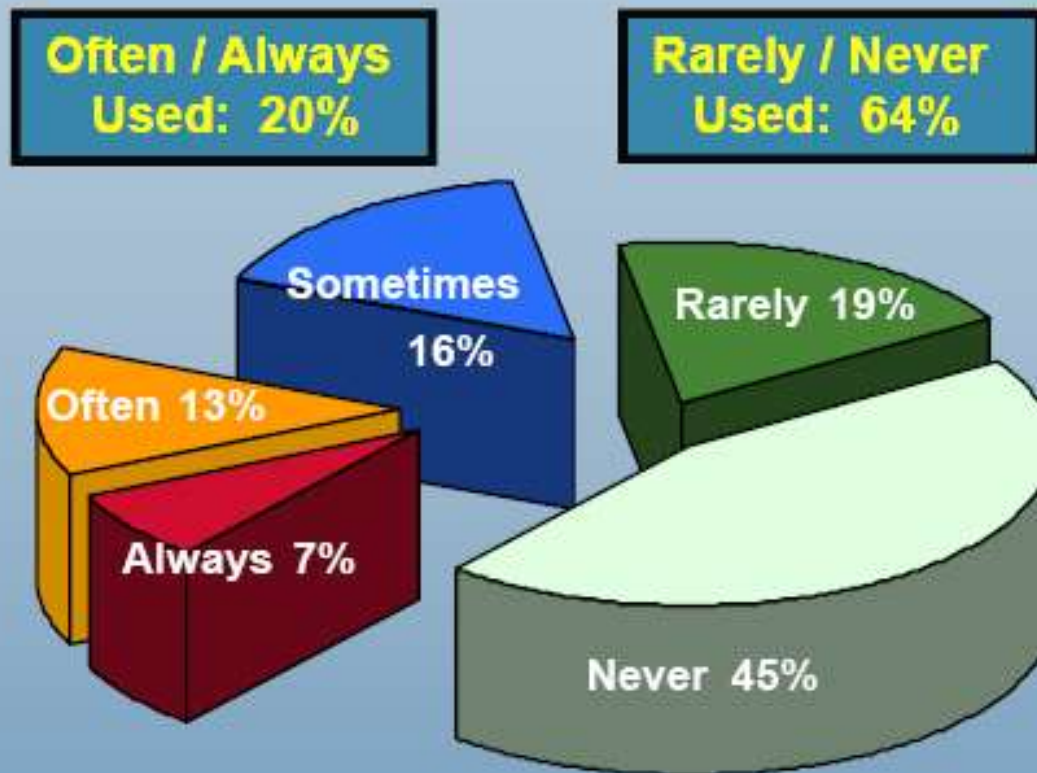
- מאפשר למשתמש חווית look & feel לקבלת משוב מוקדם על הדרישות
- מחליף דיון תיאורטי ומופשט בדרישות (IKIWISI = I Know It When I See It)



האם הדרישות אכן דרושות?

דוגמה ל-feature
המשמש לעיתים
רחוקות בלבד

Features / Functions Used in a Typical System



Standish Group Study Reported at XP2002 by Jim Johnson, Chairman

- **דרישה (Requirement)**
 - **תכונה** (property) או **יכולת** (capability) של המוצר, **הנדרשת** כדי לתת מענה לבעיה / לצורך
 - הצורך לפתור את הבעיה או להשיג את המטרה (המשתמשים)
 - הצורך לעמוד בחוזה, בתקן או במסמך מחייב אחר (בעלי עניין נוספים)
- **תכן / מימוש (Design / Implementation)**
 - פתרון (טכני) באמצעותו ניתן לענות לדרישה אחת או יותר
 - התכן בכל רמה מהווה בסיס לדרישות לרמה הבאה



דרישות פונקציונליות ודרישות לא פונקציונליות

מרחב הפתרון



דרישות לא פונקציונליות

- מגדירות מאפיינים ואילוצים על אופן

מימוש תכולת הפתרון

- מקבלות מענה כאשר הפתרון
הנבחר (התכן/המימוש) עומד
במאפיינים ובאילוצים המוגדרים

דרישות פונקציונליות

- מגדירות את תכולת הפתרון

- מקבלות מענה ספציפי וישיר בתוך
הפתרון (התכן/המימוש)

סיווג דרישות למערכות עתירות תוכנה (1)

- דרישות פונקציונאליות

– דרישות המתארות מה המערכת צריכה לעשות (באמצעות פעולות על נתונים)

- דרישה תפעולית (OR = Operational Requirement)

– דרישה המתייחסת לתפעול, לאינטראקציה או להתנהגות של המערכת

» פעולות, תרחישים, תגובות לאירועים וכו'

» פונקציות, שירותים, אלגוריתמים

» תזמונים, סדר פעולות

- דרישת מידע (DR = Data Requirement)

– דרישה המתארת ישויות מידע, ללא תלות בפעולות המתבצעות על ישויות אלה

» נתונים לגבי המערכת

» נתונים בהם המערכת מטפלת

» נתוני קלט/פלט

דרישות תפעוליות = מה המערכת צריכה לעשות
דרישות מידע = מה המערכת צריכה לדעת

דוגמאות לדרישות פונקציונליות (מהאפיון התפעולי ומהאפיון הטכני)

- **דרישות תפעוליות**

- "נוסע ... לוחץ על הכפתור ..."
- הכפתור נדלק בעקבות הלחיצה"
- "אם הטכנאי מגלה תקלה הוא מנסה לתקן אותה ולחזור על הבדיקה"

- **דרישת מידע**

- "בכל קומה יש כפתור אחד לעליה וכפתור אחד לירידה, פרט לקומה העליונה (לירידה בלבד) ולקומת הקרקע (לעליה בלבד)"

- **לתשומת לב**

- דרישות תפעוליות כוללות, בדרך כלל, גם דרישות מידע (הנתונים הרלוונטיים לפעולה)

- למשל: כפתור דלוק/כבוי

- התוכנה בדרך כלל "יודעת" פרטים אודות הישויות הפיזיות במערכת

- בייחוד אלה שעליהן מתבצעת שליטה/בקרה

סיווג דרישות למערכות עתירות תוכנה (2)

• דרישות לא פונקציונאליות – איכות הפתרון

– דרישות המגדירות תכונות נוספות של הפתרון שצריכות להתמלא תוך כדי מילוי הדרישות הפונקציונאליות (How well?)

• דרישות ביצועים (PR = Performance Requirements)

– פרמטרים ניתנים למדידה לגבי ביצועי התוכנה
» זמן תגובה, נפח איחסון, ניצולת מעבד וכו'

• מאפייני איכות (QA = Quality Attributes)

– תכונות המאפיינות את המוצר הכולל

- » אמינות (reliability) – פעולה ללא תקלות לאורך זמן
- » זמינות (availability) – שירות רצוף, התאוששות מהירה מתקלות
- » בטיחות (safety) – הגנה על המשתמשים (והסביבה) מפני המערכת
- » אבטחה (security) – הגנה על המערכת מפני משתמשים
- » בדיקתיות (testability) – היכולת לבדוק את המערכת ואת פעולותיה (גם בדיעבד)
- » אחזקתיות (maintainability) – היכולת לערוך בקלות שינויים ותיקונים במוצר
- » שימושיות (usability) – האפקטיביות והיעילות שמקנה המערכת למשתמש בביצוע משימותיו ובהשגת מטרותיו

– יש להגדיר את מאפייני האיכות באמצעות מדדים כמותיים!

סיווג דרישות למערכות עתירות תוכנה (3)

• דרישות לא פונקציונאליות – אילוצים

– דרישות ותנאים המגבילים את חופש בחירת כיווני הפתרון

• אילוץ חומרה (HC = Hardware Constraint)

– הכתבת מבנה החומרה (רכיבים, ממשקים, ארכיטקטורה), על בסיס מפרט טכני של הלקוח או תכן מערכתי ברמות גבוהות יותר

• אילוץ מימוש (IC = Implementation Constraint)

– הכתבת אופן מימוש ספציפי בתוכנה (אלגוריתם, מבנה נתונים, אופן הפעלה ספציפי, שימוש חוזר ברכיבים, שימוש בטכנולוגיה מסויימת וכו')

• אילוץ ניהולי (MC = Management Constraint)

– הכתבת תנאים ניהוליים בהם צריך לעמוד המימוש

» תקציב

» לו"ז

» זמינות משאבים

» התאמה לתקן

» ...

דוגמאות לדרישות לא פונקציונאליות (מהאפיון התפעולי ומהאפיון הטכני)

- **דרישת ביצועים**

– "מעלית כלשהי הנמצאת בכיוון הנסיעה המבוקש תגיע לקומה, תוך דקה לכל היותר "

- **דרישת בדיקתיות**

– "טכנאי, המגיע אחת ל-6 חודשים, יכול לבצע בדיקה מקיפה של כל המערכת..."

- **דרישת אחזקתיות**

– "טכנאי ... יכול לתקן תקלות..."

- **דרישת שימושיות**

– "המערכת תונגש לבעלי מוגבלויות שונות"

- **אילוץ חומרה**

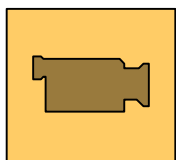
– "לכל מעלית יש בקר (מחשב) עצמאי + תוכנת בקרה למעלית"

- **אילוץ מימוש**

– "נוסע יכול לעצור את המעלית בזמן נסיעה באמצעות לחיצה על כפתור עצירת החירום"

- **אילוץ ניהולי**

– מערכת המעליות תעמוד בכל תקני הבטיחות הישימים



ניתוח אירועים – מערכות הצבעה ממוחשבות

• 2.12.2008

– בחירות מוקדמות (פריימריז) למפלגת העבודה

- התקלה: קריסת נתבי התקשורת מחמת עומס
- התוצאה: ביטול הבחירות הממוחשבות ועריכת בחירות חוזרות ידניות

• 9.12.2008

– בחירות מוקדמות (פריימריז) למפלגת הליכוד

- התקלה: המצביעים התקשו להשלים את תהליך ההצבעה ונוצרו תורים ארוכים
- התוצאה: הקלפיות נשארו פתוחות עד 01:00

• 25.11.2012

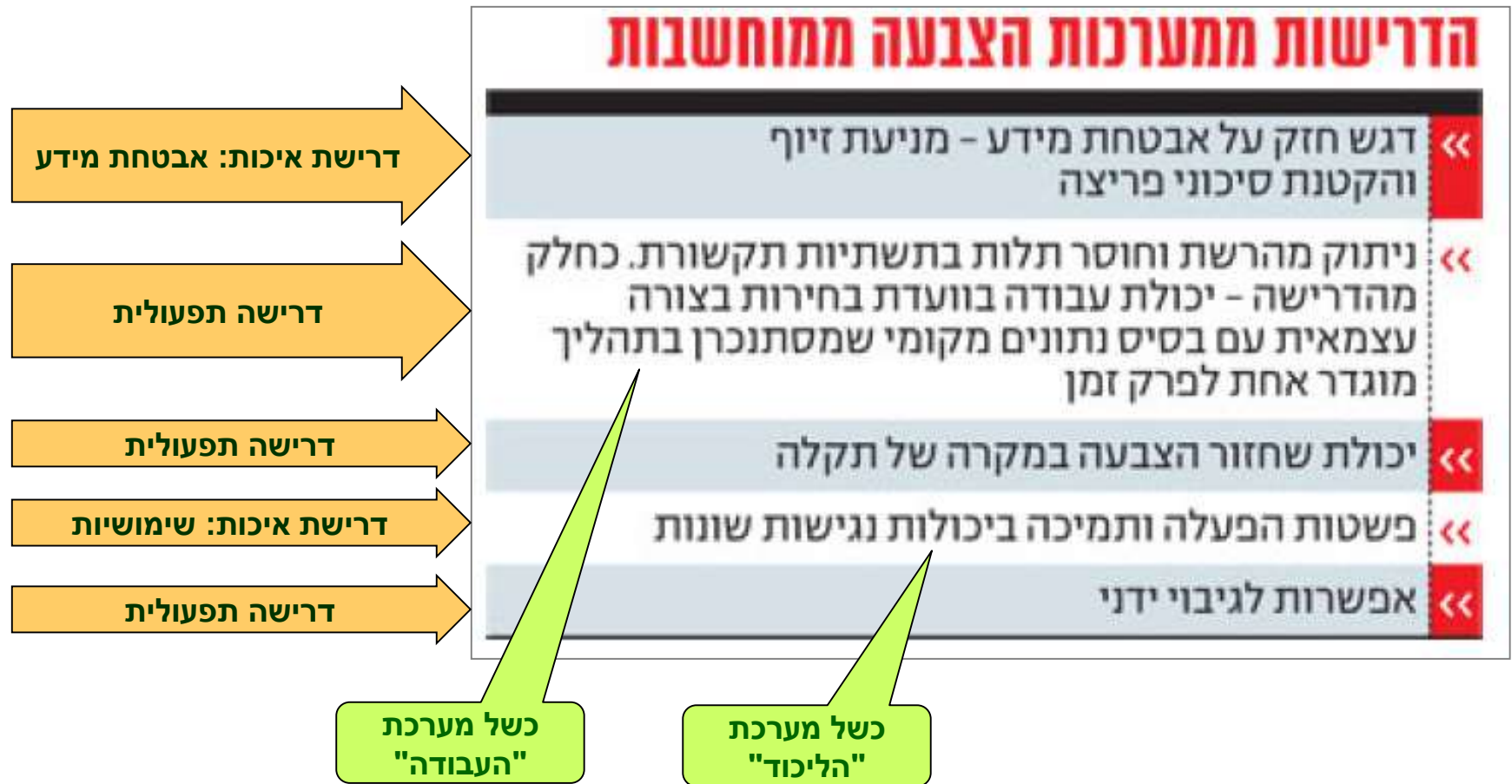
– בחירות מוקדמות (פריימריז) למפלגת הליכוד

- התקלה: תקלות שונות (קריסת שרתים, היתקעות מחשבים)
- התוצאה: המשך הבחירות ביום שלמחרת



האם מערכות ההצבעה עמדו בדרישות?

- להלן מספר דרישות פורמאליות ממערכות הצבעה ממוחשבות, אשר גובשו בועדת התקשוב העליונה במשרד האוצר [אור הירשאוגה, דה-מרקר, 11/12/08]



סינון/הפרדה של דרישות

- משפטים שמוצו מתוך המקורות הגולמיים של הדרישות עשויים להכיל מספר דרישות מסוגים שונים

- לדוגמה, המשפט הבא:

במצב "סריקה אוטומטית" המדפסת מזהה את מסמך המקור – תמונה, כרטיס ביקור או מכתב, סורקת אותו באופן אוטומטי ומאחסנת את הסריקה בזכרון, וכל זאת ללא התערבות המשתמש.

מכיל מספר דרישות, כדלהלן:

– 3 דרישות תפעוליות:

- המדפסת מזהה ...
- המדפסת סורקת ...
- המדפסת מאחסנת ...

– 4 דרישות מידע

- מצב "סריקה אוטומטית"
- מסמך מסוג "תמונה"
- מסמך מסוג "כרטיס ביקור"
- מסמך מסוג "מכתב"

– דרישת שימושיות (מאפיין איכות)

- ... ללא התערבות המשתמש

תרגילון כיתה: הפרדה של דרישות

- בכל אחד מהמשפטים הבאים זהו את הדרישות השונות, סמנו אותן ורישמו את מספרן בעמודות המתאימות

#	משפט	NFR					FR	
		MC	IC	HC	QA	PR	DR	OR
1	המערכת כוללת שרת מרכזי המספק שירות 24/7 למשתמשים המשתמשים בדפדפן מסחרי דרך תקשורת אינטרנט							
2	היישום יפעל כראוי תחת מערכות ההפעלה Windows-phone, Android, iOS							
3	המערכת תהיה מסוגלת להתאושש מקריסה בתוך 2 דקות מרגע הפעלתה מחדש, כאשר כל המידע שוחזר מתוך הגיבוי השעתי האחרון.							
4	הרכב הבלתי-מאויש דוגם את המצלמה שלו בכל 50 מילי-שניות והוא אמור להפעיל את הבלמים תוך לא יותר משניה אחת מרגע הזיהוי של מכשול בדרך.							
5	אב-טיפוס של המערכת, המבוסס על בסיס נתונים ORACLE יסופק תוך 3 חודשים מיום חתימת החוזה.							
6	לאחר זיהוי מוצלח מציג הכספומט את המסך הראשי, המכיל את האפשרויות למשיכת מזומנים, העברה מחשבון לחשבון ובירור היתרה. כפתור "ביטול" אדום ימוקם בפניה השמאלית התחתונה של המסך.							
7	מכשיר הניווט מייצר את המסלול, נותן הנחיות נסיעה וזוכר את היעדים שנבחרו בעבר. יכולת הניווט האוטומטי תסייע למשתמשים שלא ללכת לאיבוד ולהגיע ליעדיהם בקלות.							

פתרון לדוגמה לתרגילון

#	משפט	NFR					FR	
		MC	IC	HC	QA	PR	DR	OR
1	המערכת כוללת שרת מרכזי המספק שירות 24/7 למשתמשים המשתמשים בדפדפן מסחרי דרך תקשורת אינטרנט		1	2	1			
2	היישום יפעל כראוי תחת מערכות ההפעלה Windows-phone, Android, iOS		3		1			
3	המערכת תהיה מסוגלת להתאושש מקריסה בתוך 2 דקות מרגע הפעלתה מחדש, כאשר כל המידע שוחזר מתוך הגיבוי השעתי האחרון.				1		3	
4	הרכב הבלתי-מאויש דוגם את המצלמה שלו בכל 50 מילי-שניות והוא אמור להפעיל את הבלמים תוך לא יותר משניה אחת מרגע הזיהוי של מכשול בדרך.					1	1	2
5	אב-טיפוס של המערכת, המבוסס על בסיס נתונים ORACLE יסופק תוך 3 חודשים מיום חתימת החוזה.	1	1					
6	לאחר זיהוי מוצלח מציג הכספומט את המסך הראשי, המכיל את האפשרויות למשיכת מזומנים, העברה מחשבון לחשבון ובירור היתרה. כפתור "ביטול" אדום ימוקם בפניה השמאלית התחתונה של המסך.		1				4	2
7	מכשיר הניווט מייצר את המסלול, נותן הנחיות נסיעה וזוכר את היעדים שנבחרו בעבר. יכולת הניווט האוטומטי תסייע למשתמשים שלא ללכת לאיבוד ולהגיע ליעדיהם בקלות.				2		1	3

מה נדרש מהדרישות? (1)

- הדרישות צריכות להיות בדידות ומזהות (discrete and identified)

- ניתן לקרוא כל דרישה כמשפט שלם באופן עצמאי
 - לעיתים יש להפריד משפט מקורי אחד למספר משפטים
- זיהוי ייחודי וחד-ערכי לכל דרישה
 - הזיהוי נשמר לאורך כל מחזור החיים גם אם הדרישה מבוטלת!

- הדרישות צריכות להיות חד-משמעיות (unambiguous)

- לא ניתן לפרש את הדרישה באופנים שונים

- הדרישות צריכות להיות שלמות (complete)

- הדרישות מכסות היטב את כל ההיבטים ללא חסרים
 - זהירות מ-TBD (to be defined)

- הדרישות צריכות להיות עקביות (consistent)

- דרישות שונות אינן סותרות זו את זו



מה נדרש מהדרישות? (2)

- הדרישות צריכות להיות עוקבות למקור (trace to origin)
 - זיהוי מקורה של כל דרישה
 - דרישות מפורשות (explicit)
 - דרישות נגזרות (derived)
- הדרישות צריכות להימנע, ככל האפשר, מתכן (avoid design)
 - כל קביעה מפורשת של רכיב, אלגוריתם וכו' מצמצמת את מרחב האפשרויות בתכן
 - יש לוודא שאילוצי התכן / הפתרון משקפים צורך אמיתי
- הדרישות צריכות להיות בדיקות/מדידות (testable/measurable)
 - נדרשת קביעה מפורשת כיצד אפשר יהיה להוכיח את העמידה בדרישה
 - אנליזה
 - סימולציה
 - הדגמה במעבדה
 - הדגמה בהפעלה מבצעית

טבלת דרישות – הבסיס לניהול הפיתוח

מאפיינים נוספים	סוג	מקור	נוסח	זיהוי
<div> <div>...</div> <div>...</div> <div>...</div> </div>				

- זיהוי
 - חח"ע, נשמר גם כשהדרישה מבוטלת
- נוסח
 - ברור, חד משמעי, נאמן למקור
- מקור
 - בעל העניין / המסמך ממנו מוצתה הדרישה
- סוג
 - דרישה פונקציונלית / לא-פונקציונלית, לפי סוגים
- מאפיינים נוספים
 - בשקף הבא...

מאפיינים נוספים לדרישות

- **עדיפות / חשיבות ללקוח**
 - לפחות "קריטית" / "לא-קריטית"
- **רמת הקושי / סיכון טכני**
 - מידת הידע והניסיון שיש למפתח במימוש דרישה כזו
- **עלות**
 - לא בהכרח ביחס ישיר לקושי
- **אחריות / בעלות**
 - הקצאה לצוות / לתכנת / לשותף / לקבלן משנה וכו'
- **רציונל: למה זה נדרש? כיצד נקבעו הערכים?**
 - בעיקר למניעת השקעה מיותרת הנובעת מ"מפרט-יתר" (over specification)
- **אופן ההוכחה / הבדיקה**
 - דמיות (similarity) / ניתוח / סימולציה / בדיקה במעבדה / ניסוי או פיילוט / תפעול בשטח
- **סטטוס**
 - הדרישה הוצעה / נבחנה / נותחה / הוקצתה למרכיב פתרון / מומשה / נבדקה
- **... (מאפיינים נוספים על פי הצורך)**

טבלת דרישות (חלקית) למערכת המעליות*

NFR	FR	נוסח	זיהוי
HC	DR	המערכת כוללת 3 מעליות	1
	DR	המערכת משרתת 10 קומות	2
HC	DR	בכל קומה, פרט לקומת הקרקע ולקומה העליונה, יהיו שני כפתורים – אחד לעליה ואחד לירידה	3
HC	DR	בקומת הקרקע יהיה כפתור אחד לעליה בלבד, בקומה העליונה כפתור אחד לירידה בלבד	4
	OR	נוסע הנמצא בקומה כלשהי ... לוחץ על הכפתור המתאים לכיוון הנסיעה המבוקש	5
<div>▪ ▪ ▪</div>			
HC		במעלית נמצא כפתור לעצירת חירום	
HC		במעלית נמצא כפתור להזעקת חילוץ	
<div>▪ ▪ ▪</div>			
QA		המערכת תונגש לבעלי מוגבלויות שונות	

מטלה: בניית טבלת דרישות

- מתוך "סיפור הלקוח ובעלי העניין" של ePark יש למצות דרישות לתוך טבלת אקסל ולסווג אותן
 - יש לשמור ככל האפשר על הניסוח המקורי
 - מומלץ לעבוד על המסמך המקורי ע"י "שבירת שורות" ואחר כך להעביר את הטקסט לטבלה
 - יש להשלים טקסט על מנת שאפשר יהיה לקרוא כל דרישה באופן עצמאי, גם כאשר היא עומדת בנפרד מהדרישות האחרות
 - יש להשתמש בתבנית Requirements-Table-Template.xlsx
- שמרו את טבלת הדרישות שבניתם בתיקיה האישית

ניהול דרישות בכלים ממוחשבים

- ניהול הדרישות לאורך מחזור החיים יכול להתבצע בצורה אפקטיבית באמצעות טבלה המכילה דרישות ומאפיינים

- קיימים כלים ממוחשבים (לדוגמה Doors של IBM), המאפשרים, בדרך כלל

- יצירת טבלאות היררכיות של דרישות
- יצירת קישורים בין דרישות
- יצירת עקיבות (traceability) בין הדרישות למסמכי המקור
- יצירת עקיבות מהדרישות לתוצרי התכן
- גזירת טבלת אימות ותיקוף (V&V) לצורך ניהול בדיקות וניסויים
- יצירת מפרטי דרישות טקסטואליים
- ניהול גרסאות של הדרישות לאורך מחזור החיים

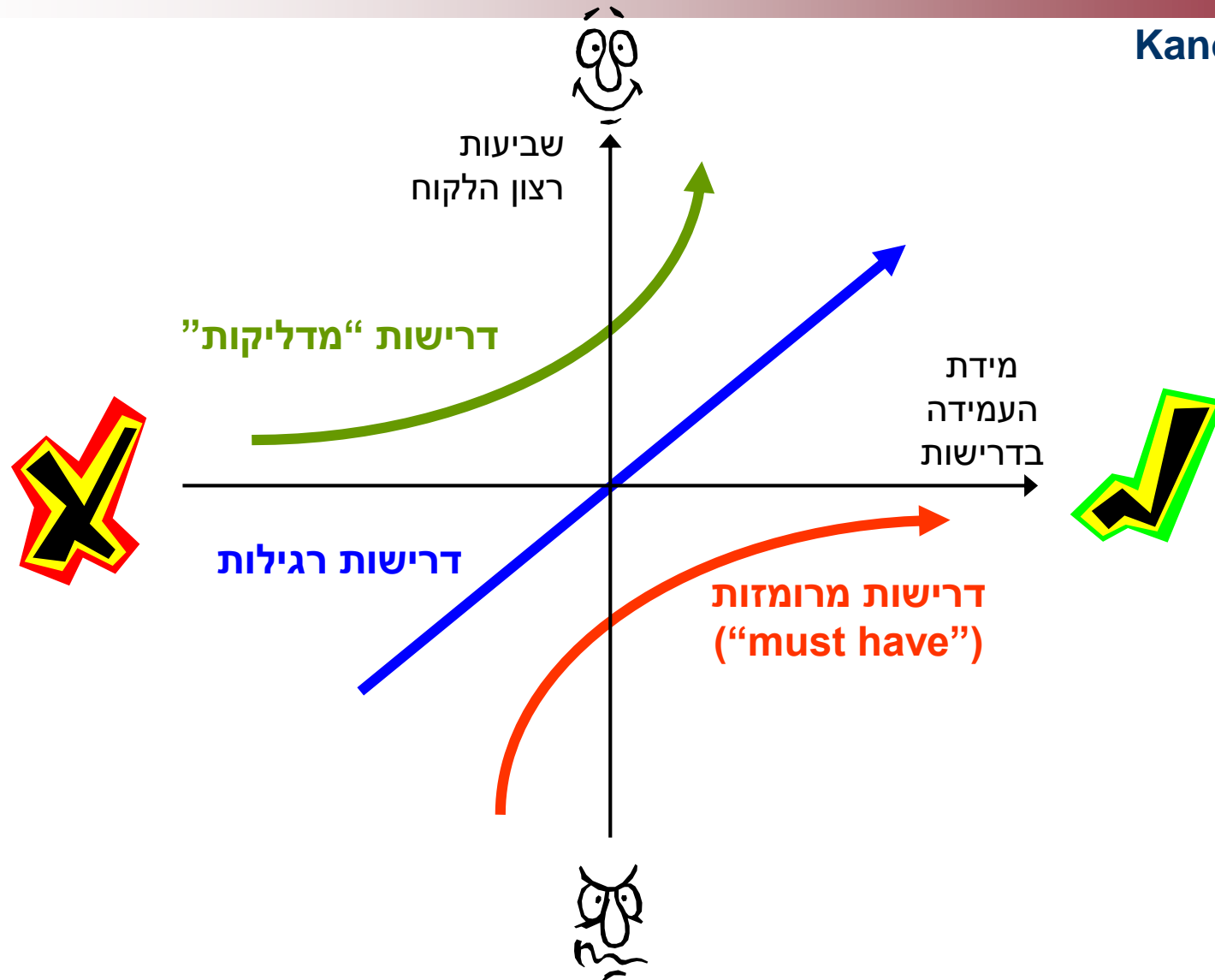
ID	Requirements in Volera Template	Use Cases to Link by Name or ID	Conflicting Parts	Dependee Parts
R-12	1 ATM Requirements			
R-11	1.1 Customer Services			
R-3	Customer can withdraw cash	Withdraw some Cash		R-39
R-2	Customer can obtain statements	Obtain a Statement		R-39
R-14	Customer can order a chequebook	Order a Chequebook		R-39
R-15	Customer can change password	Change Password		R-39 R-43
R-18	1.2 Utilities			
R-39	ATM can authenticate the customer	UC-19 UC-18 UC-32 UC-13		
R-41	ATM can obtain current balance	UC-36		
R-45	ATM can count out cash	UC-37		
R-47	ATM can ensure cash includes small notes	UC-37		
R-46	ATM can dispense cash	UC-38		
R-42	ATM can debit customer's account	UC-39		
R-43	ATM can update customer's password	UC-40		
R-40	1.3 Customer NFRs			
R-44	All transactions must be quick and simple			
R-16	Customer accounts must be extra secure	Change Password Withdraw some Cash Obtain a Statement	R-44	R-39
R-17	Cash withdrawal must be very reliable	Withdraw some Cash		R-3

דוגמה לניהול דרישות ב-Doors

http://www.scenarioplus.org.uk/papers/traceability/trace_to_uc.htm

דרישות נסתרות

מודל Kano



דוגמאות לדרישות נסתרות



- דרישות מרומזות (tacit requirements)

- ניתן יהיה להפעיל לפחות אחת מהמעליות במתכונת "שבת"
[דרישה פונקציונלית מבעל עניין נוסף]
- המעלית תעצור באופן "חלק" ולא בפתאומיות [בטיחות]

- דרישות "מדליקות" (exciting requirements)

- בכל קומה יהיה שעון דיגיטלי המציג את זמן ההמתנה למעלית
הבאה [דרישה פונקציונלית + אילוץ תכן]
- ניתן יהיה להפעיל את המעלית באמצעות הסמארטפון [שימושיות]

דרישות איכות – דרישות נסתרות

- אפיונים של מערכת הינם בדרך אפיונים פונקציונאליים

- הדרישות הלא-פונקציונאליות (מאפייני האיכות) הן דרישות נסתרות, אשר אי-קיומן עלול להתגלות רק בשלבים מאוחרים, ולגרום אי-שביעות-רצון למשתמשים

- "למה זה עולה כל כך לאט?"

- "למה זה נתקע לעיתים קרובות?"

- "מחקתי בטעות מידע חשוב, האם אפשר לשחזר אותו?"

- אפשר לגלות חלק מהדרישות הנסתרות ע"י שאלות שאלות של "how well" לגבי המאפיינים הפונקציונאליים

- לדוגמה, נתונה הדרישה הפונקציונאלית:

- "אפליקציית הניווט תשווה את מיקום הרכב, כפי שהתקבל מה-GPS, עם המסלול שעל גבי המפה, ובמקרה של סטייה מהמסלול יחושב מסלול מחדש"

- שאלות "how well":

- "מה קורה כאשר ה-GPS לא זמין?"

- "מהו דיוק ההשוואה?"

- "האם המשתמש יכול לבטל את חישוב המסלול מחדש?"

- ...

- שאלות אלה יביאו לגזירה/הוספה של דרישות לא-פונקציונאליות ופונקציונאליות נוספות

- כאשר ה-GPS לא זמין במשך TBD שניות תתריע האפליקציה ע"י ...

- כאשר ה-GPS אינו זמין האפליקציה תחשב את המיקום המשוער של הרכב ע"י חיתוך בין מיקומן של אנטנות סלולריות

- **דרישה פונקציונאלית**

– הקצאת הדרישה למרכיב של התכן/המימוש אשר יספק בפועל את הפונקציונליות הנדרשת

- דוגמה: אלגוריתם, מאגר נתונים, פונקציה

- **דרישה לא פונקציונאלית**

– בחירת חלופת תכן/מימוש אשר ניתן להוכיח כי יישומה יביא לעמידה בדרישה, תוך כדי אספקת הפונקציונליות הנדרשת

- דוגמה 1: בחירת אלגוריתם חסכני בזמן/במקום, כדי לעמוד בדרישת ביצועים

- דוגמה 2: בחירת ארכיטקטורה של שרתים כפולים כדי לעמוד בדרישת זמינות

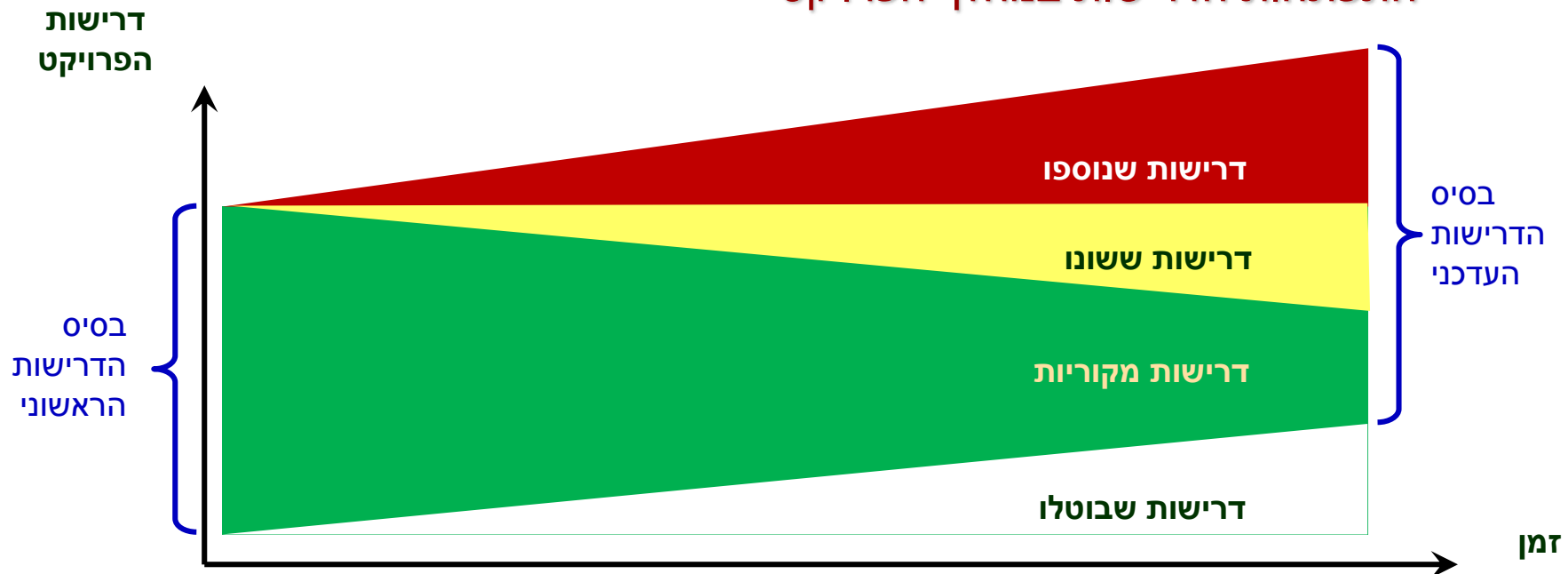
– גזירת דרישות פונקציונליות, אשר מימושן יביא לעמידה בדרישה

- דוגמה 1: הכנסת דרישות של הצפנת מידע על מנת לענות לדרישת בטחון

- דוגמה 2: הכנסת דרישות תפעוליות מפורטות להבטחת שימושיות ראויה

התפתחות הדרישות במהלך הפרויקט

- בדרך כלל הדרישות אינן נשארות קבועות במהלך הפרויקט
 - דרישות עשויות להתווסף, להשתנות או להתבטל
 - לתופעה זו השלכות משני סוגים
- השלכות הנדסיות – שינויים בתכן ובמימוש המוצר כתוצאה משינויים בדרישות
- השלכות ניהוליות – שינויים בתוכניות הפרויקט עקב השתנות בסיס הדרישות
- התפתחות הדרישות במהלך הפרויקט*



*מקור: Systems Engineering Handbook, INCOSE, 2010