

תהליך התכן ההנדסי של מערכת

מרצה: ד"ר דרורה גושן-מסקין

מפגש רביעי – 21.7.13

פתוח התכן

1. תזכורת – תהליך התכן ההנדסי:
 - מיצוי דרישות
 - פתוח דרישות
2. פתוח (דרישות) התכן:
 - 2.1 תהליך פתוח התכן המערכתי
 - 2.2 החלטות תכן מערכתיות
 - 2.3 הקצאת משימות, מאפיינים טכניים וקונספט תפעול למכלולים
 - 2.4 הקצאת דרישות לכל מכלול
3. השלמת בסיס המידע ההנדסי
4. אימות ותיקוף התכן ההנדסי
5. תהליך התכן ההנדסי או הנדסת דרישות

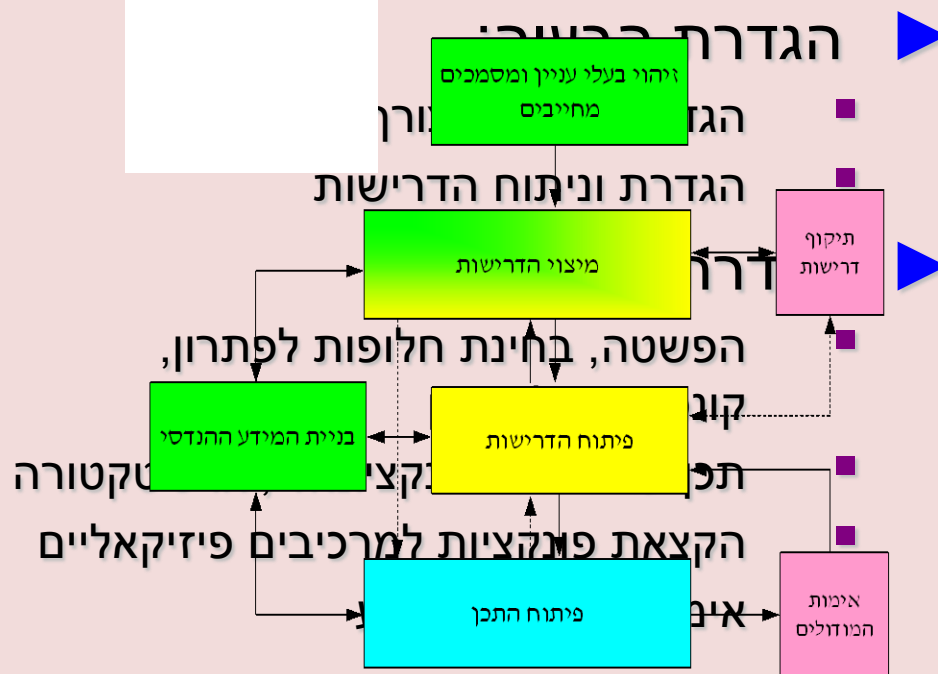
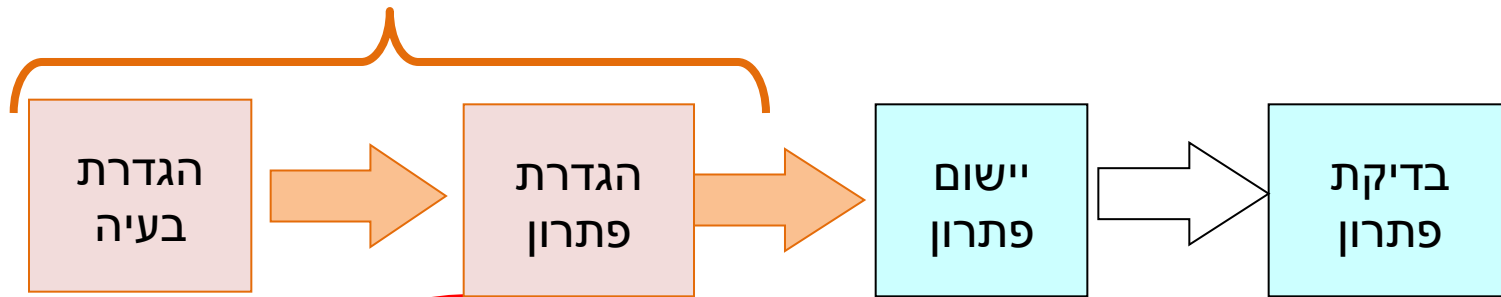
נספח - בניית תכניות עבודה

פרק 1

תזכורת – תהליך התכן ההנדסי

תהליך הפתוח המערכתי

תהליך התכנהנדסי



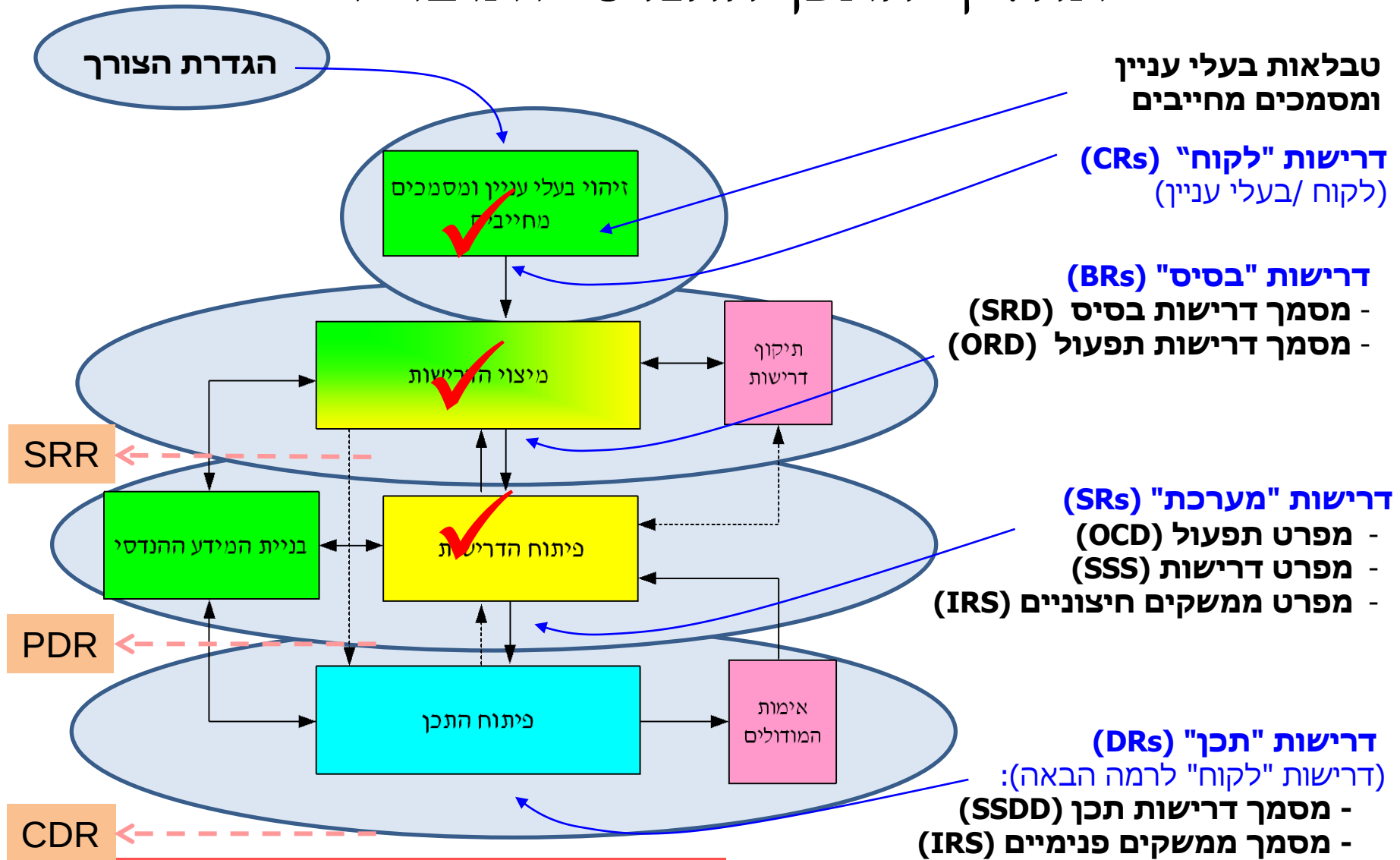
יישום/בניית הפתרון:

- תכנהפורט של כל מרכיב
- בניית מרכיבים
- בדיקת מרכיבים

בדיקת הפתרון הכולל:

- V & V מערכתי
- T & E למערכת

תהליך התכן ההנדסי ותוצריו



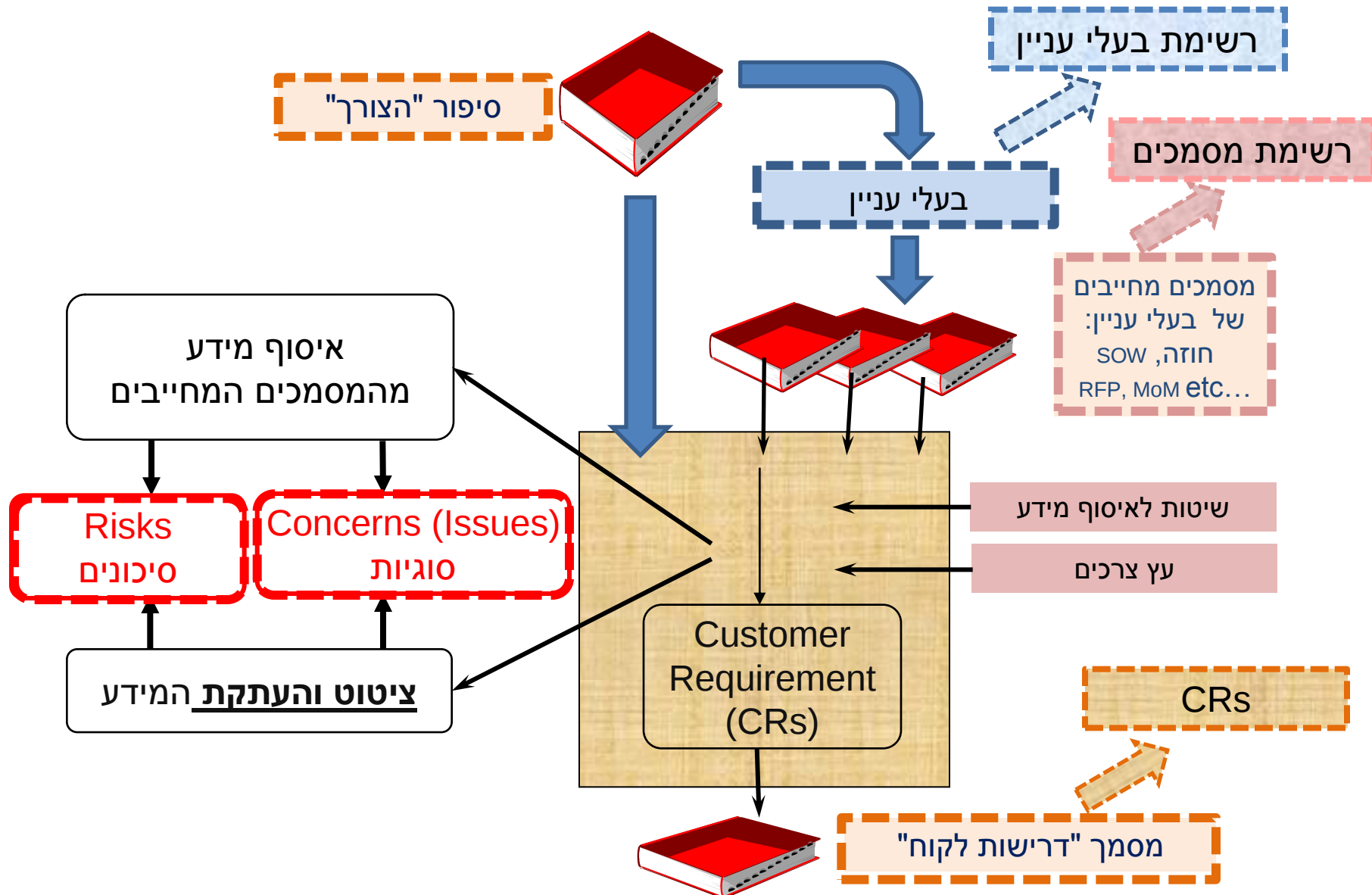
דרישות בסיס חתומות – תחת נוהל שינויים רשמי

דרישות מערכת חתומות – תחת נוהל שינויים רשמי

כל הזכויות שמורות לרשות המרכזית למידע ומידע

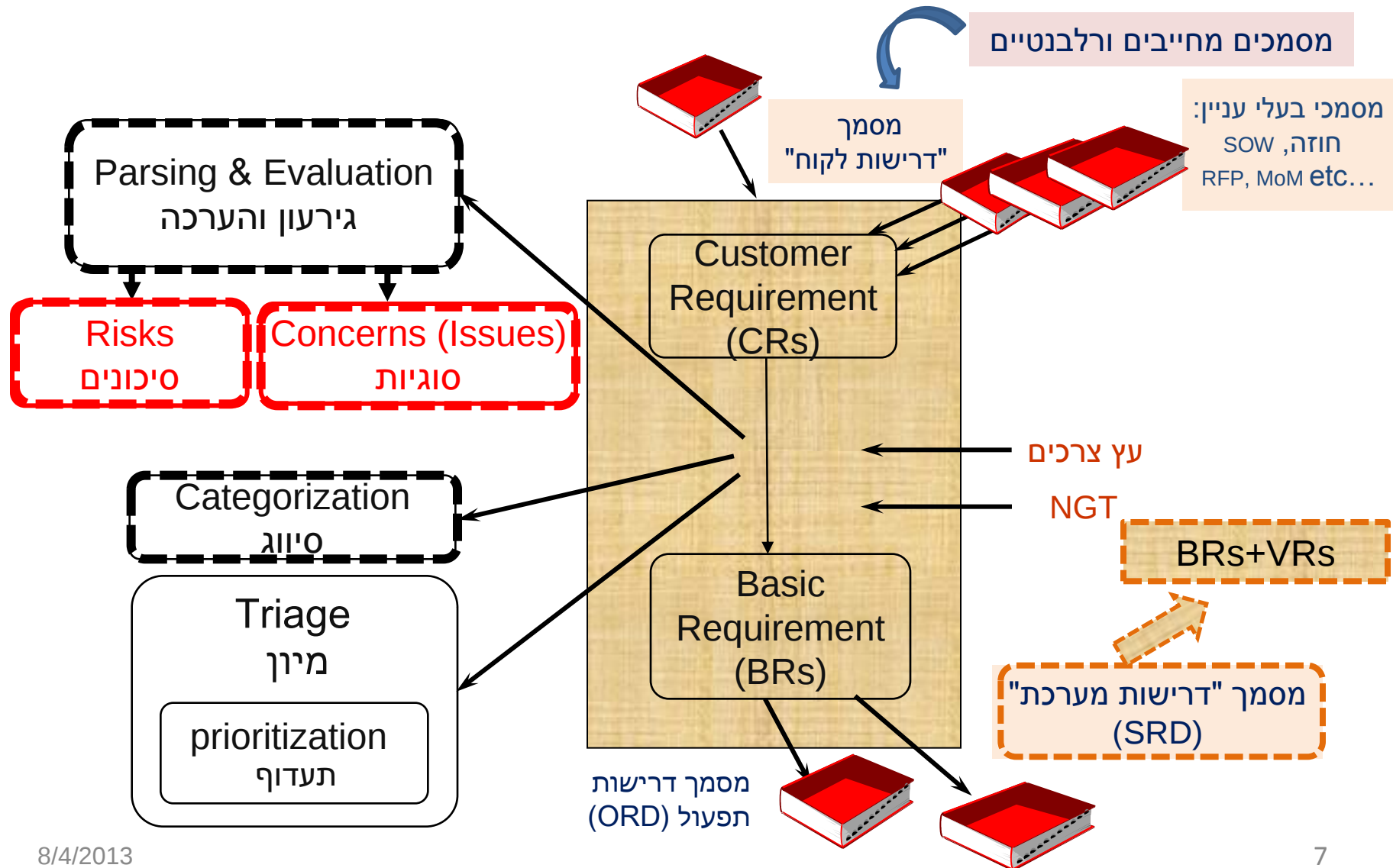
שלב א – זיהוי בעלי עניין ומסמכים מחייבים

מסיפור הצורך לדרישות לקוח



שלב ב' - מיצוי דרישות

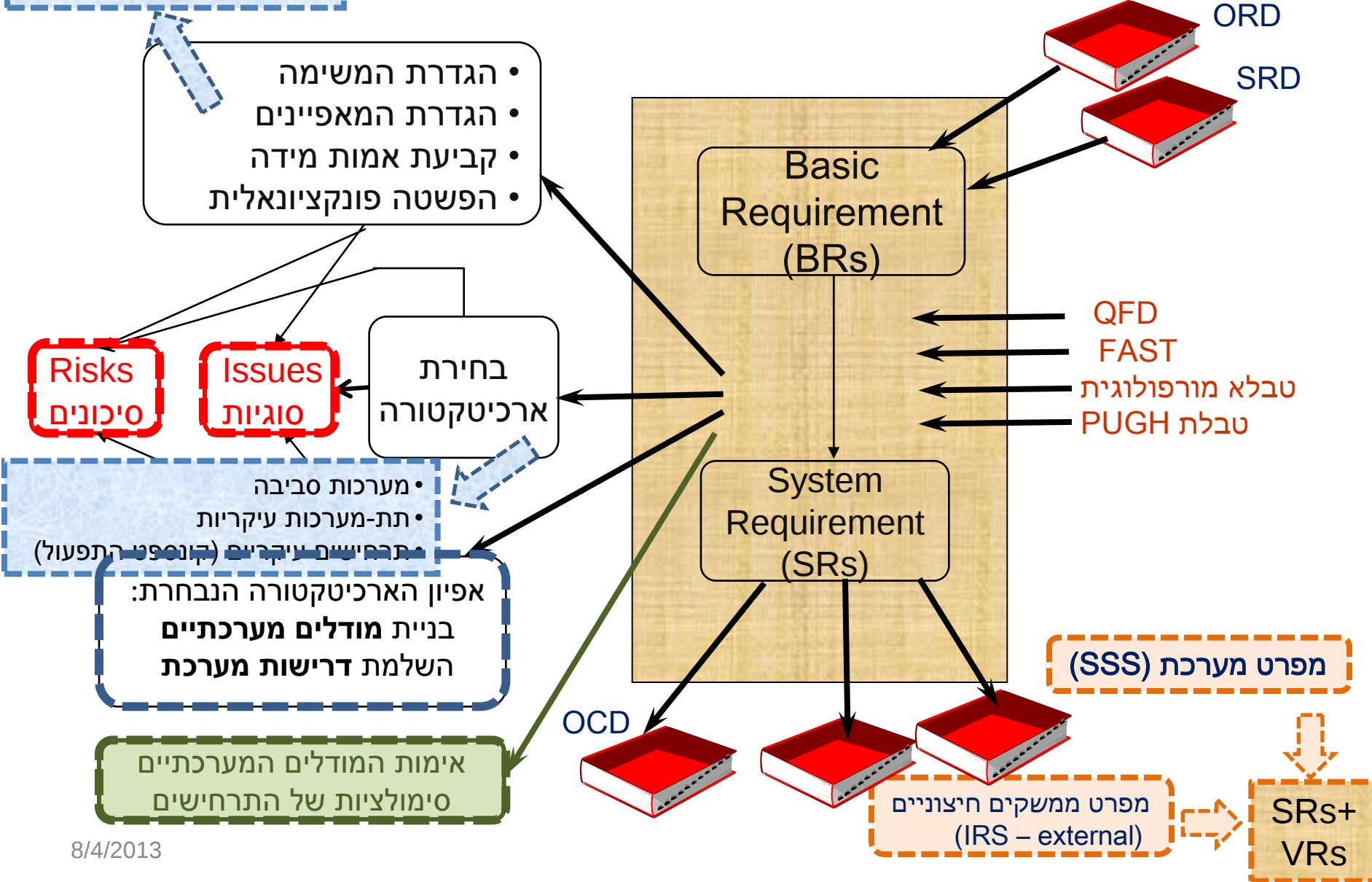
מדרישות לקוח (צורך) לדרישות בסיס



שלב ג' - פתוח דרישות

מדרישות בסיס לדרישות מערכת

- דרישות מערכת עיקריות
- פונקציות סביבה
- פונקציות מערכתיות עיקריות



שלב ד' - פתוח התכנ

מדרישות מערכת לדרישות תכנ (מערכת)

תוספת לדרישות
המערכת העיקריות

החלטות תכנ מערכתיות

עבור כל מכלול:

- הגדרת משימה
- הגדרת מאפיינים
- הגדרת קונספט תפעול

Risks
סיכונים

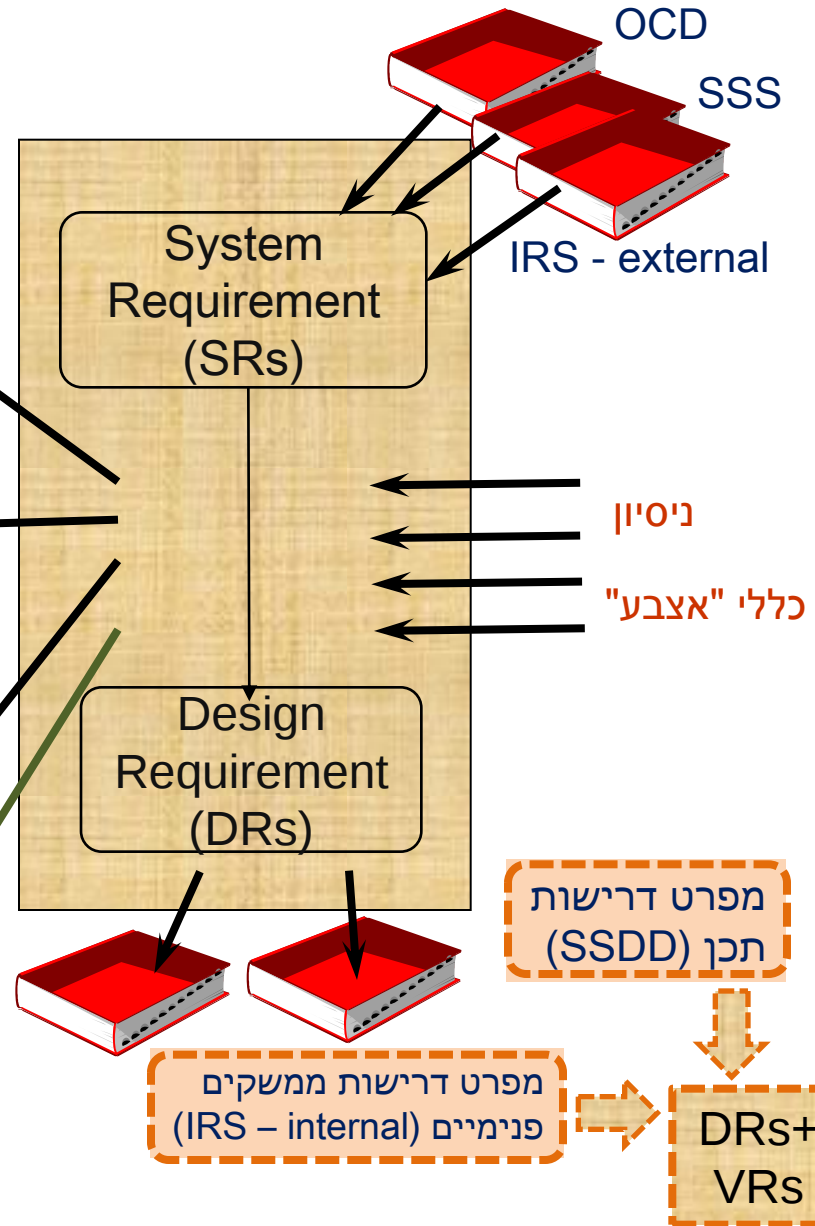
Issues
סוגיות

אפיון כל מכלול:

- הקצאת פונקציות
- הקצאת דרישות פונקציונליות
- הקצאת דרישות מאלצות
- הגדרת תרחישי על

- דרישות מכלול עיקריות
- פונקציות סביבת מכלול
- פונקציות מכלול עיקריות
- מערכות סביבת מכלול
- תרחישים עיקריים

אימות התכנ – סימולציות של התרחישים



ה"קפיצה" הראשונה

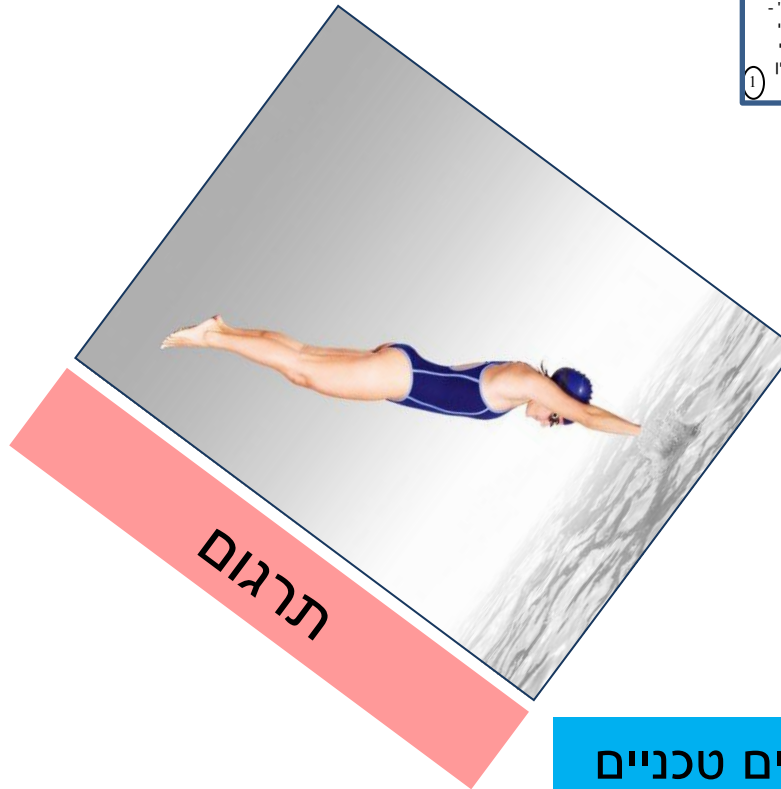


דרישות בסיס של הלקוח

- איכותיים
- מבטאים את ה"מה"

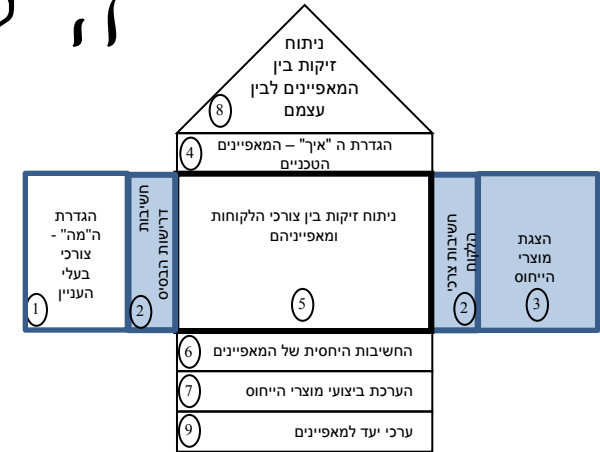


הגדרת המשימה

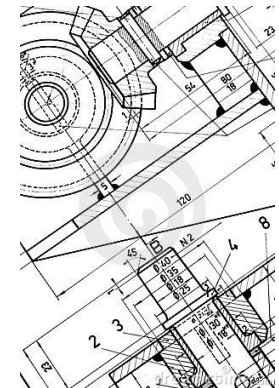


מאפיינים טכניים של המערכת

- מדידים
- מבטאים ביצועים טכניים נדרשים כדי לבצע את ה"מה"



Criteria selection



ה"קפיצה" השנייה

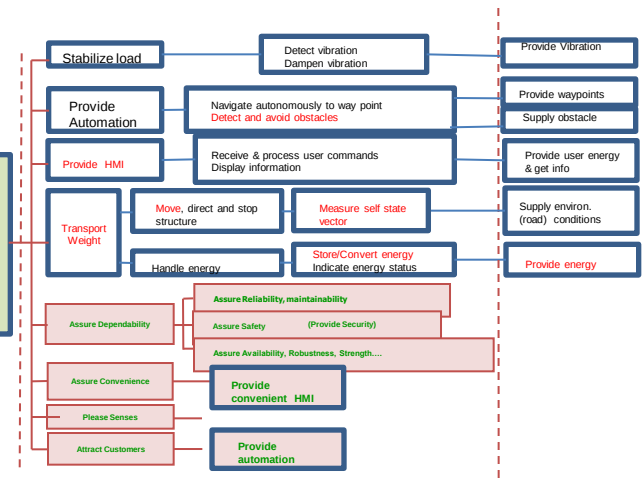


דרישות הבסיס
+
משימת המערכת
+
מאפיינים טכניים

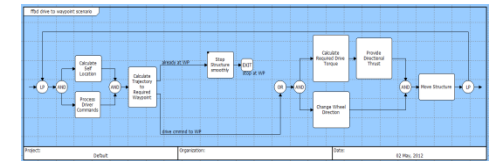
הפחטה פונקציונאלית



Provide affordable self mobility capability at nearby neighborhood



	J=3	J=2	J=1	
		P_{12} 5;3	P_{11} 5;5	F_1
P_{ij} 3;5		P_{ij} 3	P_{21} 5;5	F_2
	P_{33} 5;3	P_{32} 3;5	P_{31} 3;5	F_3



פונקציות (תפקודי) המערכת

- ביטויים (כלליים, מופשטים ונטולי פתרון) הכוללים **פועל ושם עצם** ומתארים "מה המערכת עושה?"
- **רצוי** שהעצם יתאר תכונה ברת מדידה



בניית מודל קונספטואלי/פיסיקאלי למערכת

בחירת חלופה למערכת
(טבלה מורפולוגית)

קונספט התפעול המערכתי

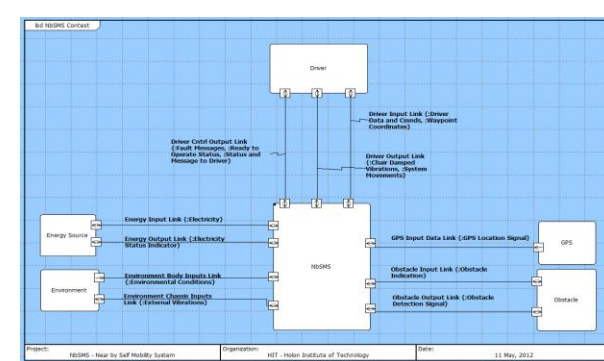
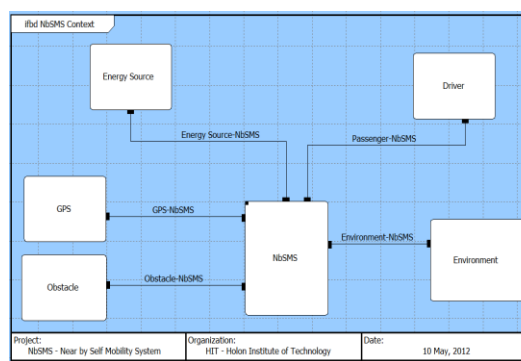
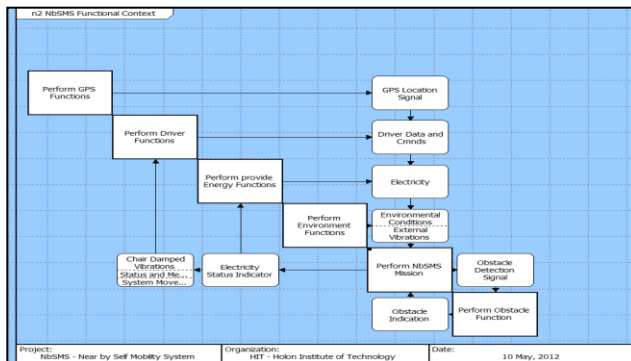
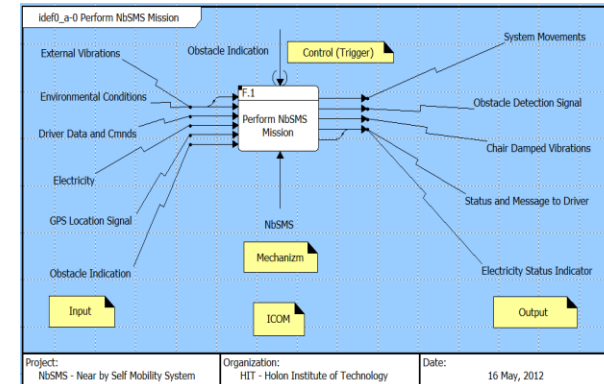
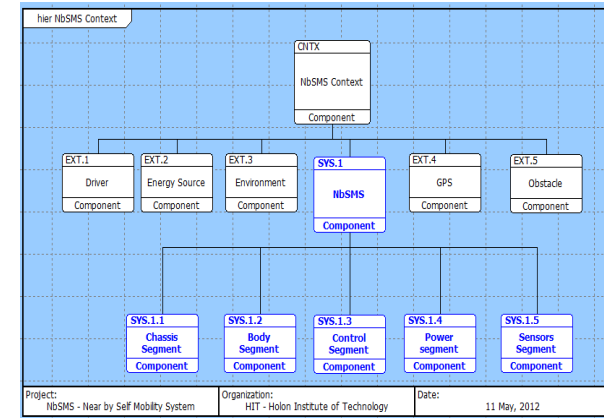
תרשים הארכיטקטורה
הראשית

תרשים המערכת בסביבתה
(External Context View)

תרשים הארכיטקטורה
(Internal Context View)

תרשים ממשקים חיצוניים (N^2)

המודל הפיסיקלי/
קונספטואלי



בנית מודל פונקציונלי למערכת

הפשטה של המערכת
(תרישים FAST)

תרחישי התפעול המערכתיים

פונקציות/תפקודים עיקריים

עץ פונקציות מערכתיות



תהליכים דינמיים



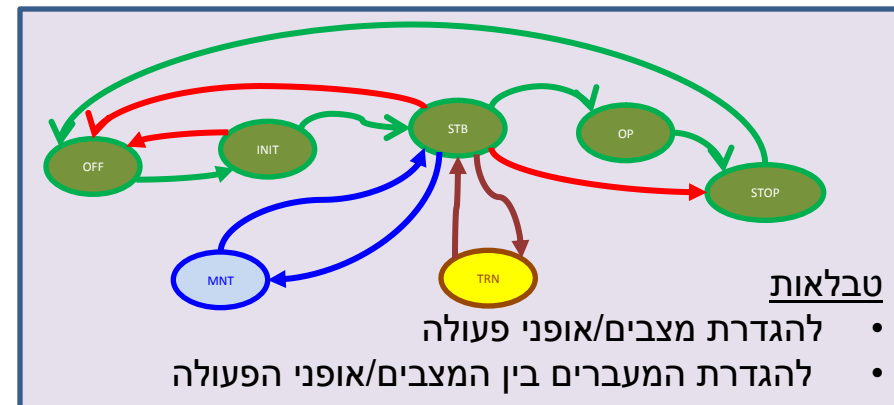
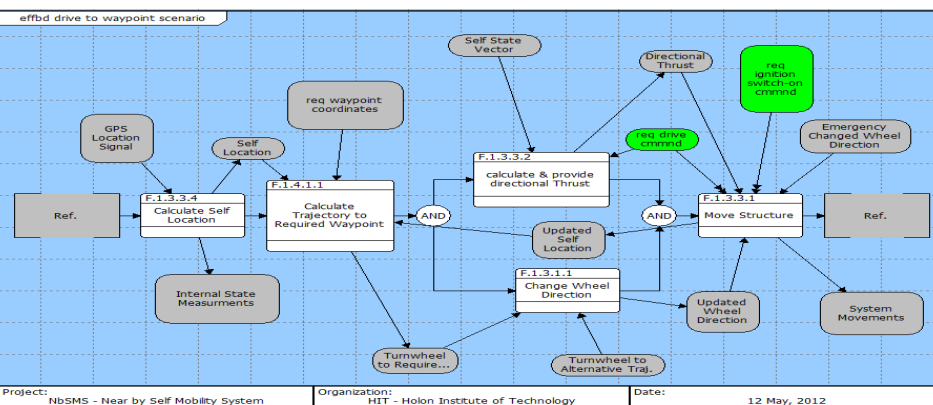
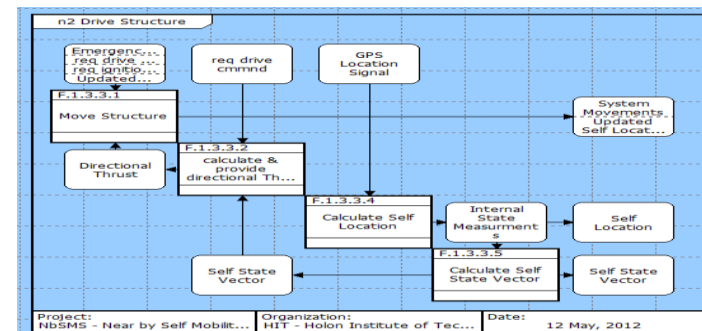
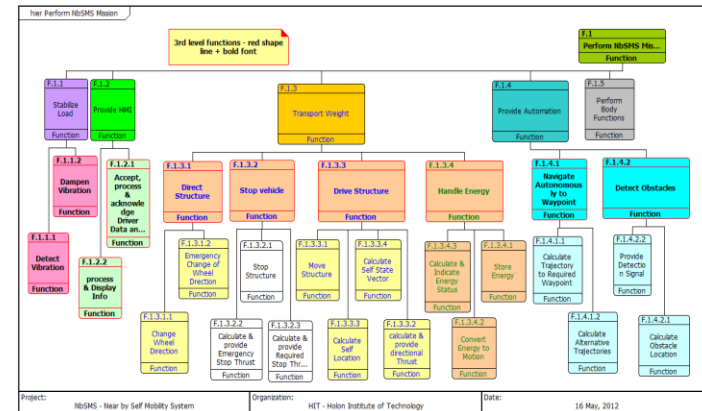
(תרחישים הכוללים זרימות קלט/פלט)



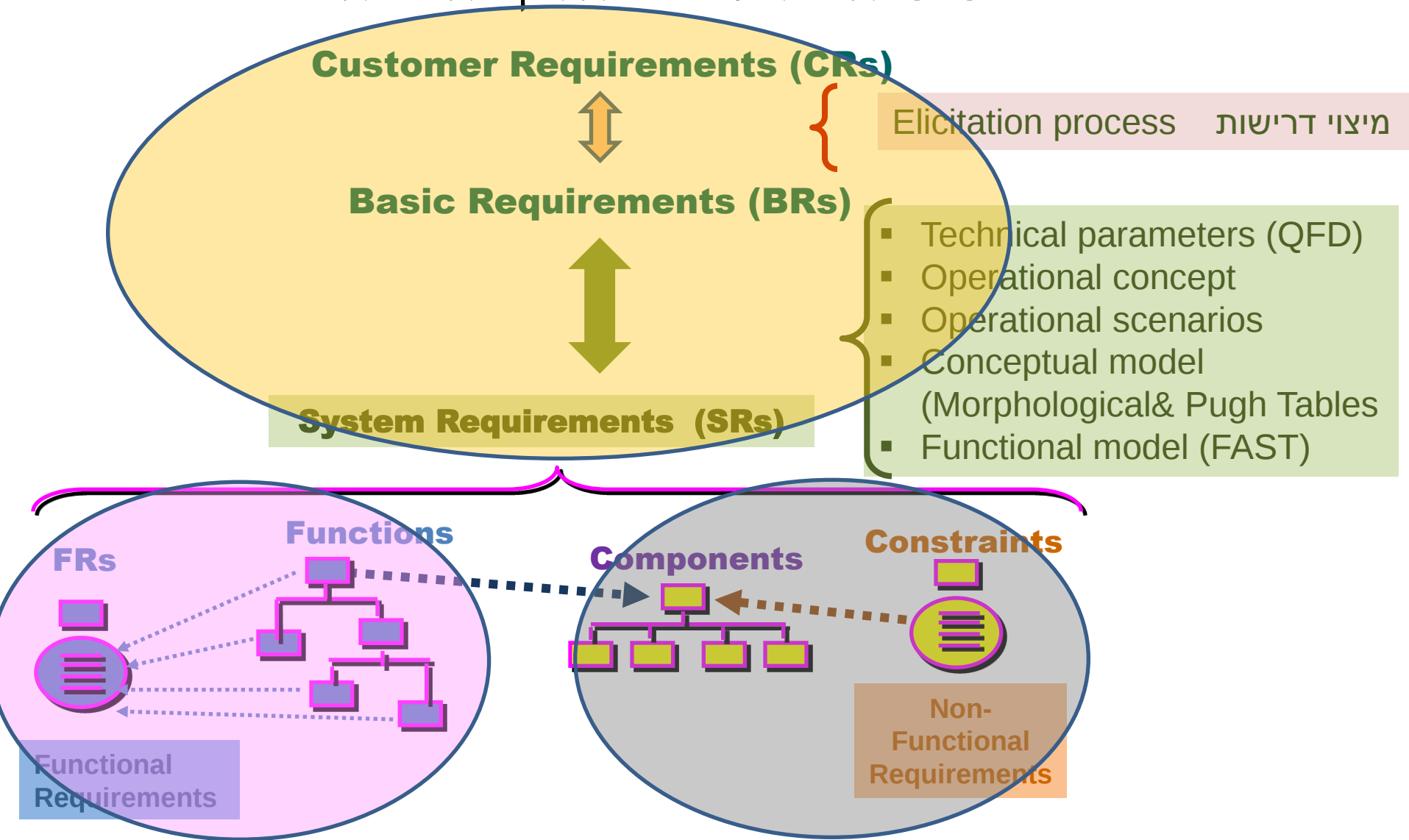
תרחישי ממשקים פנימיים (בין פונקציות)

אופני פעולה/מצבי מערכת/מעברים

המודל הפונקציונלי
(ההתנהגות המערכתית)



בסיס המידע – תהליך הבניה



המטרה: התאמה בין ההתנהגות לבין הארכיטקטורה

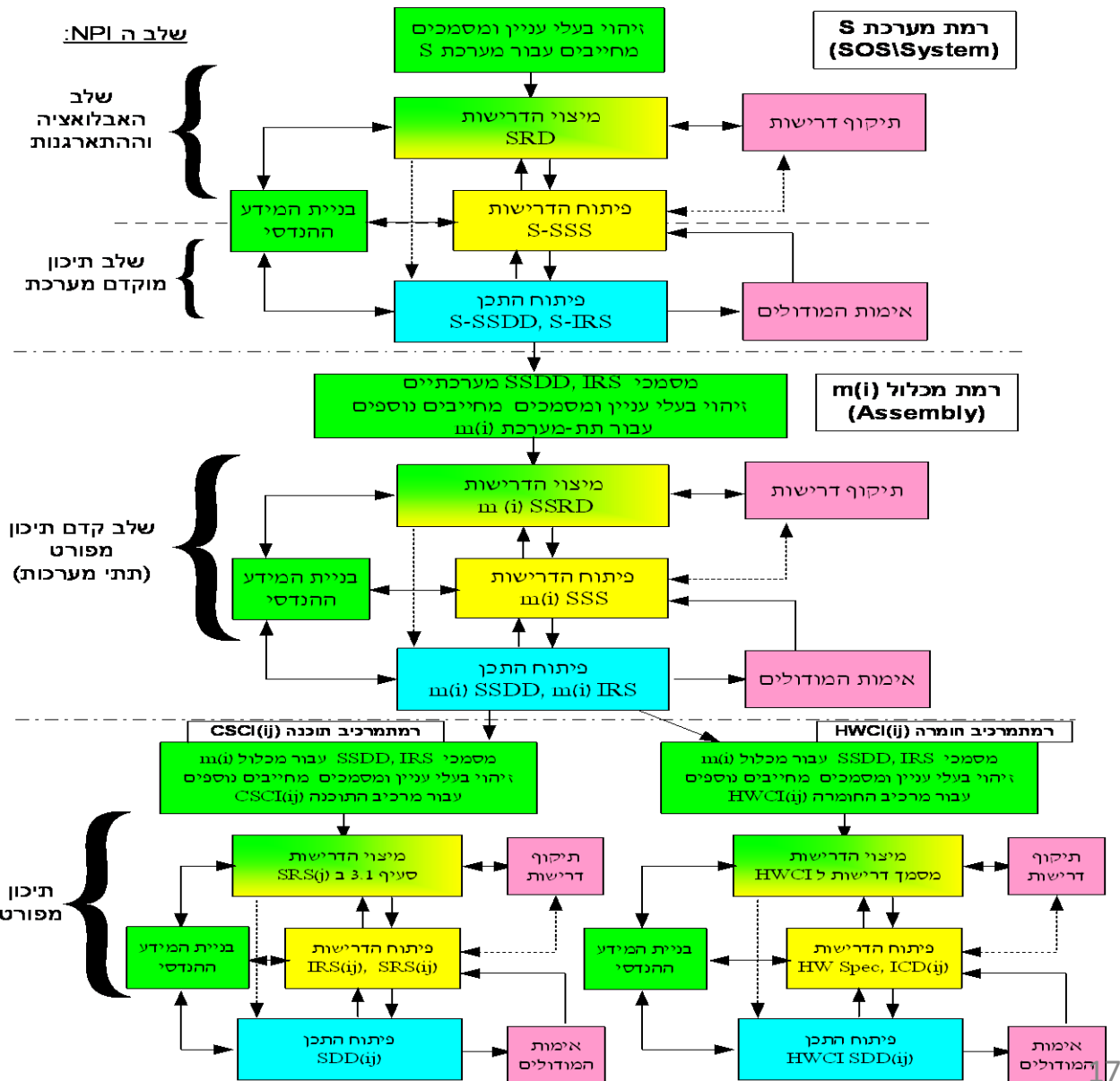
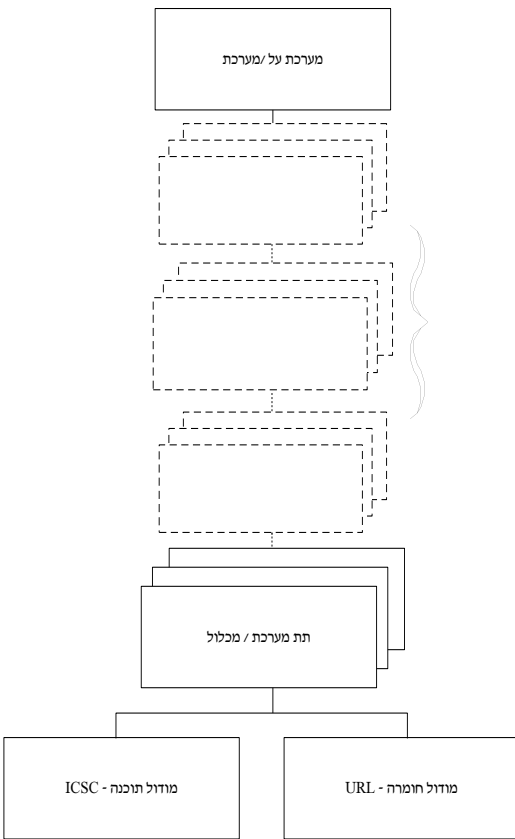
תפוקות ותוצרים ל- PDR מערכת

מסמכים רשמיים בסיום תהליך פיתוח הדרישות	תוצרי תהליך פיתוח הדרישות
מבוססי מודלים • מפרט דרישות (SSS) • מפרט ממשקים (IRS) • קונספט התפעול המבצעי (OCD) • קונספט התפעול לאורך מחזור החיים (LCOCD)	• בית האיכות להגדרת מאפיינים (QFD) • אמות המידה להערכת החלופות • טבלאות מורפולוגיות • טבלאות הבחירה של החלופות (Pugh) • מנגנוני כשל וחישובי אמינות (CFMA) FTA • טבלאות לעלויות יעד • סיכונים ותוכניות להורדת סיכונים • תוכניות עבודה • תוצאות בדיקות האימות השונות • בסיס נתונים מעודכן

פרק 2.1

תהליך פתוח התכן המערכת

תהליך תכן הנדסי במערכת



העברת התכן למהנדסי יישום (חומרה/תוכנה)



8/4/2013

פתוח התכן המערכתי - המטרה

הגדרת "דרישות לקוח" (*) "לכל מכלול (פיסיקלי/קונספטואלי)

ניסוח ותיעוד דרישות התכן המערכתיות (SSDD)
ניסוח ותיעוד דרישות ממשקים פנימיים (Internal IRS)



הצגת מסמכי התכן ב- CDR

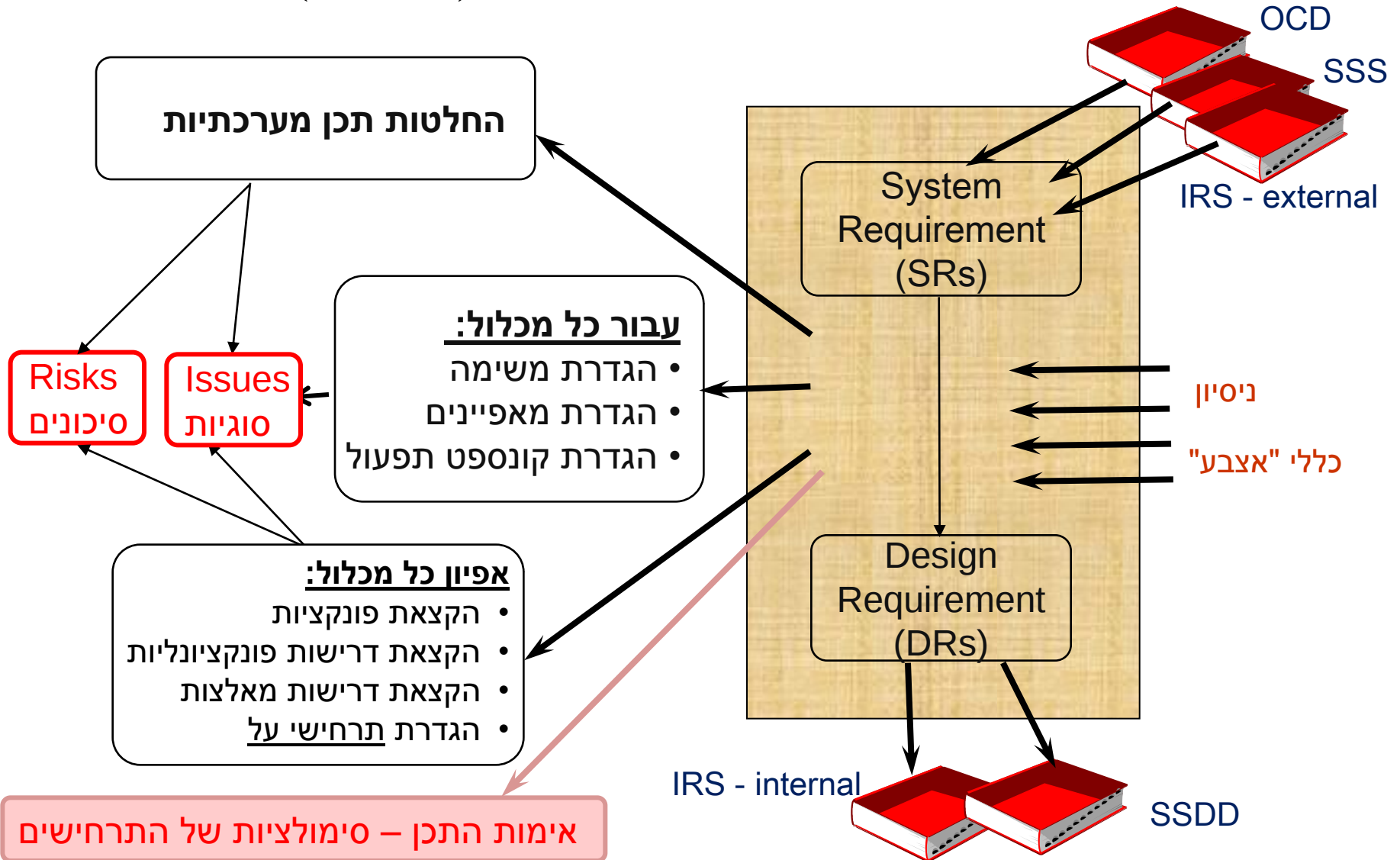
(*) – "דרישות לקוח" אלה יכולים להיות גם
"דרישות בסיס":

- אם אין בעלי עניין אחרים
- אם כתובים בצורה נכונה (לפי הכללים)

יש להיזהר שמסמכי הקצאת
הדרישות לא יהפכו
ל"דרישות מערכת" של
המכלול (כלומר שלא יהוו את
מסמכי ה- SSS של
המכלולים (קומפוננטות)

פתוח התכנ

מדרישות מערכת לדרישות תכנ (מערכת)



הגדרת/הקצאת "דרישות לקוח" לכל מכלול (פיסיקלי/קונספטואלי)

- דרישות ייצור והרכבה
- דרישות שימוש במרכיבים קיימים
- דרישות שילוב ובדיקות
- דרישות אריזה, אחסון, תובלה ושינוע
- דרישות מחזור/גריסה
- דרישות RAMS:
 - תחזוקה ו- ILS
 - אמינות, זמינות ובטיחות
- דרישות תיעוד והדרכה
- דרישות חוסן, חסינות ואבטחה,
- דרישות תנאי סביבה
- דרישות איכות:
 - הוכחת כושר/אימות ותיקוף
 - שמירת איכות הסביבה
- השלמה ובדיקת בסיסי הנתונים

- החלטות תכן מערכתיות
- הקצאת משימות
- הקצאת מאפיינים טכניים
- הגדרת קונספט התפעול
- קביעת ארכיטקטורה
- הקצאת פונקציות עיקריות
- הגדרת תרחישים עיקריים
- השלמת בסיס המידע

- דרישות לעלויות (לו"ז)
- דרישות ממשקים
- דרישות לעיצוב המוצר
- דרישות לכושר גידול

תהליך ההקצאה

- קבלת החלטות תכן מערכתיות (System Wide Design Decisions)
- הגדרת משימות ומאפיינים טכניים **לכל מכלול** (= תתי-מערכות = קומפוננטות)
- הגדרת קונספט התפעול **לכל מכלול**
- קביעת ארכיטקטורה מערכתית (הגדרת כל "מכלול בסביבתו")
- **הקצאת הפונקציות למכלולים תוך הגדרת הממשקים החיצוניים**
- **הגדרת תרחישים לכל מכלול תוך שימוש בפונקציות המוקצות עבורו**
- **השלמת בסיס המידע** $\leq$ קבלת דרישות תכן (DRs) **לכל מכלול:**
 - **הקצאת** $SRs (= Constraints + FRs)$ **למכלולים** והפיכתם ל- DRs
 - **הגדרת דרישות תכן חדשות**
 - **קישור** דו-כיווני של ה- DRs ל- SRs

Architecture is the way in which one assigns the functional elements of a product to its physical chunks and how these physical chunks interact to achieve the product's overall function.

PDMA Glossary for New Product Development

פרק 2.2

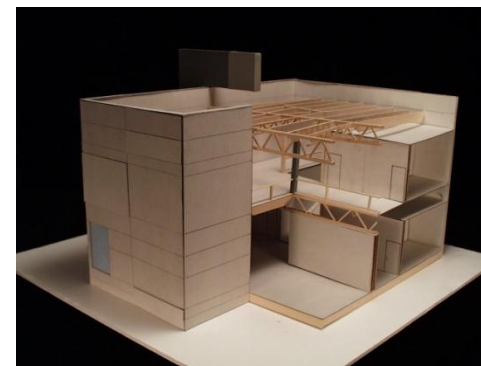
החלטות תכן מערכות

ה"קפיצה" השלישית



דרישות מערכת

פתוח התכן:
הקצאת דרישות

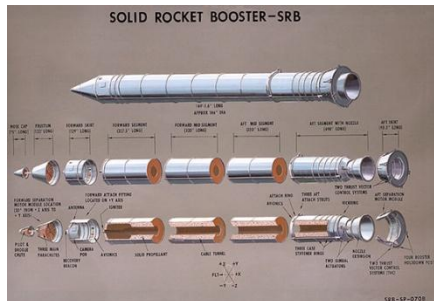


החלטות תכן מערכתיות

Solid Rocket Booster



רוחב פסי הרכבת בארה"ב הוא 143.5 ס"מ



"As an accident of history most road carriages in the Middle Ages inherited the old Roman cart gauge of approximately 4 feet, 8-1/2 inches. Julius Caesar set this width under Roman law so that vehicles could traverse Roman villages and towns without getting caught in stone ruts of differing widths. Over the centuries this became the traditional standard". the book *Gordian Knot: Political Gridlock on the Information Highway* by W. Russell Neuman, Lee McKnight, and Richard Jay Solomon.



החלטות תכן מערכתיות (System Wide Design Decisions)

- מטרה: קבלת מכלולים ברי שליטה וניהול שיבטיחו הצלחת הפרויקט
 - החלטות מוקדמות מדי
 - החלטות מאוחרות מדי
- ← שינויים מהותיים
פרויקט לא מנוהל
- החלטות התכן המערכתיות מתקבלות בתנאי אי-וודאות אך צריכות להיות מבוססות עד כמה שאפשר על:
 - סימולציות
 - Trade studies
 - עקרון מנחה – תלות מינימאלית וברת שליטה בין המכלולים
 - החלטות התכן המערכתיות מוסיפות אלמנטים למודלים המערכתיים. עשויות גם לשנות אלמנטים.
 - המלצה: סקרי עמיתים לקבלת מידע מפרויקטים קודמים/אחרים

החלטות תכן מערכתיות - דוגמאות

- חלוקת מטלות בין מפעיל למערכת
 - מפעיל מתכנן, המערכת מבצעת תוך בקרה תמידית
 - מערכת אוטונומית לחלוטין
- מימוש בתוכנה או בחומרה
 - מהירות תגובה נדרשת לעומת מהירות חישוב זמינה
 - גמישות לשינויים עתידיים בתוכנה מול שדרוגי חומרה
- מערכת מודולארית מול מערכת אינטגרלית:
 - חישוב מרכזי או מבוזר
 - מערכת מודולארית מאפשרת שינויים וכושר גידול עתידי
 - מערכת אינטגרלית טובה יותר בביצועים (בד"כ) ועולה פחות
- בחירת ממשקים:
 - ממשקים פיסיים
 - ממשקי תקשורת
- כושר גידול
 - כושר גידול יכול להיות יקר (במיוחד אם בסופו של דבר לא ינוצל)
 - כושר גידול יכול להיות מנוף לשיפור היכולת השיווקית לאורך מחזור החיים

תכן ארכיטקטורה:
חלוקה לתתי-מערכות /
מכלולים

דוגמא

הקצאת פונקצית הבקרה והניווט במל"ט

בקרה ושליטה על המטוס דוגמא לאוטונומיות גדלה

- בקרה ישירה של ההגאים
- בקרת זוויות ההגאים
- בקרת זוויות מטוס (עלרוד, גלגול, אזימוט)
- בקרת מסלול מטוס
- בקרת מיקום מטוס
- בקרת משימה



טבלת Fitts המקורית (1951)

היבטים בהם מכונות נוכחיות עדיפות על אנשים

- יכולת תגובה מהירה לאותות בקרה ויכולת הפעלת כוח רב באופן חלק ומדויק.
- יכולת לבצע פעילויות שגרתיות חוזרות.
- יכולת לאחסן מידע לזמן קצר מאד ואז למחוק אותו לחלוטין.
- יכולת ניתוח, הסקה וחישוב דדוקטיביים (מהכלל אל הפרט).
- יכולת לבצע פעילות מורכבת מאד, למשל לבצע דברים רבים בעת ובעונה אחת.

היבטים בהם אנשים עדיפים על מכונות נוכחיות

- יכולת להבחין באנרגיה קולית או חזותית קטנה.
- יכולת להבחין בתבניות של אותות אור או קול.
- יכולת לאלתר ולהשתמש בנהלים גמישים.
- יכולת לאחסן מידע רב לזמן ממושך ולהחזיר נתונים רלבנטיים כשהם נדרשים.
- יכולת להסיק מסקנות מהפרט אל הכלל.
- יכולת להפעיל שיפוט.

מיון של חלוקות אחריות בין אדם ומחשב

1. האדם מתכנן הכול ומעביר למחשב את המשימה לביצוע דטרמיניסטי.
2. המחשב מציג חלופות. האדם בוחר ביניהן, מתכנן פעילויות ומעביר את המשימה למחשב לבצע.
3. המחשב מסייע בקביעת חלופות וממליץ על החלופה העדיפה. האדם בוחר בין החלופות ומעביר את המשימה למחשב לבצע.
4. המחשב בוחר חלופה ומתכנן פעולה. האדם מאשר או דוחה. המחשב יכול להשתמש בחלופות שהוצעו ע"י ידי האדם בעבר.
5. המחשב בוחר פעולות ומבצע אותן אם האדם מאשר.
6. המחשב בוחר חלופות, תוכניות ופעולות ומציג אותן בזמן כך שהאדם יכול להתערב. בהעדר התערבות, מבצע המחשב לבדו כברירת מחדל.
7. המחשב מבצע את המשימה כולה ומדווח לאדם מה בוצע.
8. המחשב מבצע את כל המשימה ומדווח לאדם רק אם התבקש לכך.
9. המחשב מבצע את כל המשימה ומדווח רק אם מאמין שהאדם זקוק לכך.
10. המחשב מבצע אוטונומית את כל המשימה ומתעלם בכלל מהאדם. המפעיל חייב להאמין למחשב בכל ההיבטים של קבלת החלטות.

מערכת מודולארית/אינטגרלית

חישוב מבוזר

- מחשבים נפרדים בחלק ממכלולי המערכת.
- העברת מידע בין המחשבים:



– Shared Memory

– Message Passing

Server/main
computer?

טופולוגית תקשורת:

- כוכב
- טבעת (ring)
- Bus
- Mesh

ארכיטקטורת תקשורת:

- RS232
- TCP/IP
- ETHERNET
- CAN-BUS

חישוב מרכזי

- מחשב מרכזי באחד ממכלולי המערכת – מבצע את כל פעולות החישוב והבקרה עבור כלל המכלולים.
- העברת מידע בתוך המחשב בין תהליכים שונים:

– Shared Memory

– Message Passing

שיקולים לתכן ארכיטקטורה (מכלולים)

- תכן ארכיטקטורה תפקידו לקבוע את התמונה הכוללת של המכלולים המרכיבים את המערכת, כולל הבקרה והממשקים ביניהם.
- מתודולוגיות תכן קובעות את עקרונות תכן הארכיטקטורה: **מודולאריות, הפשטה והסתרת מידע**
- חלוקת מערכת למכלולים מתבצעת בהסתמך על עקרונות התכן שמהם נובעים המושגים הבאים:
 - **מיקוד** (cohesion) – התמקדות כל מכלול בנושא ספציפי אחד.
 - **צימוד** (coupling) – יחסי גומלין בין מכלולים (פונקציונאליים/לא-פונקציונאליים).
 - **סיבוכיות** (complexity) – מורכבות מערכתית
- שיקולים נוספים לתכן:
 - הפחתת סיכונים – ע"י "חלוקת" גורמי סיכון בין מכלולים או ריכוזם במכלול אחד
 - רובוסטיות לשינויים פיסיקליים
 - זיהוי קומפוננטות הנוטות לשינויים
 - זיהוי קומפוננטות העומדות בפני שינוי טכנולוגי
 - "הגיון" פנימי אחיד לביצוע ההקצאה.
 - התאמה למבנה ארגוני (קבלני משנה וכו').

מתודולוגיות תכן תוכנה/מערכת

מערכת

- **מיקוד** פונקציונאלי של כל מכלול
- אי-תלות פונקציונאלית
- מספר מינימאלי של ממשקים בין מכלולים
- ממשקים פשוטים ולא מסובכים (מינימום סינכרון בין קלטים/פלטים, וכו')

- תהליך התכן ההנדסי זהה לתהליך ההפשטה בתוכנה.

- **כימוס** (encapsulation) של הפונקציות במכלול
- הקטנת ה**צימוד** בין המכלולים

תוכנה

מודולריות (Modularity):

תוכנה מחולקת לאלמנטים נפרדים של פונקציות קשירות ובלתי תלויות, שנקראים מודולים (Modules). לכל מודול שם וכתובת משלו. חשיבות החלוקה למודולים נובעת מהעובדה שהמורכבות הכוללת של בעיה גדולה מסכום המורכבויות של מרכיביה, וכן מאמץ/זמן הפתרון של בעיה בכללותה גדול מסכום המאמצים/זמנים לפתרון מרכיביה.

הפשטה (Abstraction):

הפשטה היא התרכזות בבעיה ברמה של הכללה מבלי להתייחס לפרטים בלתי רלוונטיים ברמה נמוכה יותר. בכל שלב של הנדסת התוכנה קיימת רמה שונה של הפשטה. בשלב הניתוח רמת ההפשטה היא הגבוהה ביותר והיא מתעדנת כשעוברים לשלבי תוכן-על ותכן מפורט עד שמגיעים בקוד לרמת ההפשטה הנמוכה ביותר.

הסתרת מידע (Information Hiding):

עיקרון הסתרת המידע מאפשר חלוקת מערכת למודולים. ע"פ העיקרון כל מודול מסתיר את המידע שבו מכל המודולים האחרים ומוסר להם רק את הנתונים הדרושים להם.

גישות לאופטימיזציה של התכן

- אופטימיזציה של תתי-מערכות

- מינימום צימוד
- מקסימום מיקוד
- מינימום סיבוכיות



אבל: צימוד
מינימאלי מוריד
סיבוכיות, מאפשר
גמישות, אבל עלול
לפגוע בביצועים!!

- אופטימיזציה מערכתית

- מיקוד פונקציונאלי מקסימאלי
- מספר מינימאלי של תתי-מערכות
- תתי-מערכות בעלי יציבות פנימית
- צימוד מידע לפי הנדרש להשגת אופטימיזציה מערכתית.

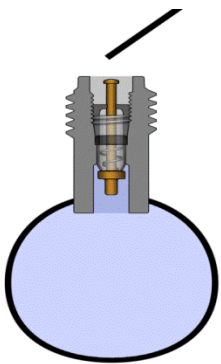


המטרה:
**אופטימיזציה
מערכתית**

מינימום צימוד = הקטנת מספר הממשקים

- עיקר עבודתם של מהנדסי מערכת זוטרים בפרויקט הוא הגדרת, תיכון ובדיקת ממשקים בין מרכיבי המערכת.
- ממשקים הינם מקור לתקלות; ככל שיש יותר ממשקים בין מרכיבי המערכת כך התיכון מסובך יותר.
- ממשק הינו מקור לטעויות תיכון העלולות לפגוע באינטגרציה/שילוב מרכיבים ובכך לפגוע בפעולת המערכת.
- פחות ממשקים = פחות מקורות לטעויות, סיבוכים, תקלות וקשיי אינטגרציה.
- עיקרון תיכון חשוב בהנדסת מערכות:

כמה שפחות ממשקים יותר טוב



ממשקים – החלטת תכן מערכתית

- ממשקים טובים הינם:

- יציבים – בדרך כלל אינם משתנים לאורך חיי הפרויקט
- סטנדרטיים – מאפשרים תיאום והבנה הדדית מהירים
- בעלי חוסן (Robust) – מאפשרים כושר גידול (רוחב סרט, מספר פינים, מרחב פיסי)

- ממשקי תקשורת מאופיינים בדרישות ביצועים של:

- תפוקה - Throughput
- זמן תגובה - Response Time
- מהימנות - Fidelity

- סטנדרטים של ממשקי תקשורת ותוכנה:

- מודל 7 השכבות של OSI
- הסטנדרט TCP/IP
- CORBA – הסטנדרט התעשייתי למערכות תוכנה

מבנה הממשק מגדיר:

- פונקציות לניהול הממשק
- פריטי קלט/פלט

הרחבת המודל
הפונקציונאלי
(ההתנהגות
המערכתית)

OSI - Open System
Interconnection
TCP/IP – Transport Control
Protocol/Internet Protocol
CORBA - Common Object
Request Broker architecture)

תהליך מומלץ לתכן מערכת

- שרטוט דיאגרמת N^2 (פונקציונלית)
- זיהוי "קבוצות" של פונקציות בעלות צימוד חזק במיוחד (לפי הקשרים בין הפונקציות).
- שרטוט דיאגרמת N^2 לפי ה"קבוצות" הפונקציונליות:
 - בחינת מידת הצימוד בין ה"קבוצות"
 - בחינת המיקוד שהתקבל
 - בחינת מידת היציבות הפנימית של ה"קבוצות"
- הערה: יש כאן מקום למחקר....

N² chart representation

A	0		0	0	0					Output
0	B	0	0	0	0					↔
	0	C	0				0	0		
0	0	0	D				0	0		
0	0			E	0	0				↑
0	0				F	0	0			↓
		0	0	0	0	G	0			
		0	0		0	0	H			

8

March 2011

Applying holisitic thinking / 1.21



N² chart representation

A	0		0	0	0				
0	B	0	0	0	0				
	0	C	0				0	0	
0	0	0	D				0	0	
0	0			E	0	0			
0	0			0	F	0	0		
		0	0	0	0	G	0		
		0	0		0	0	H		

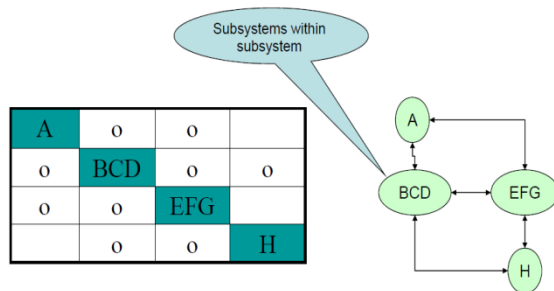
9

March 2011

Applying holisitic thinking / 1.21



Aggregated (synthesized) subsystems



10

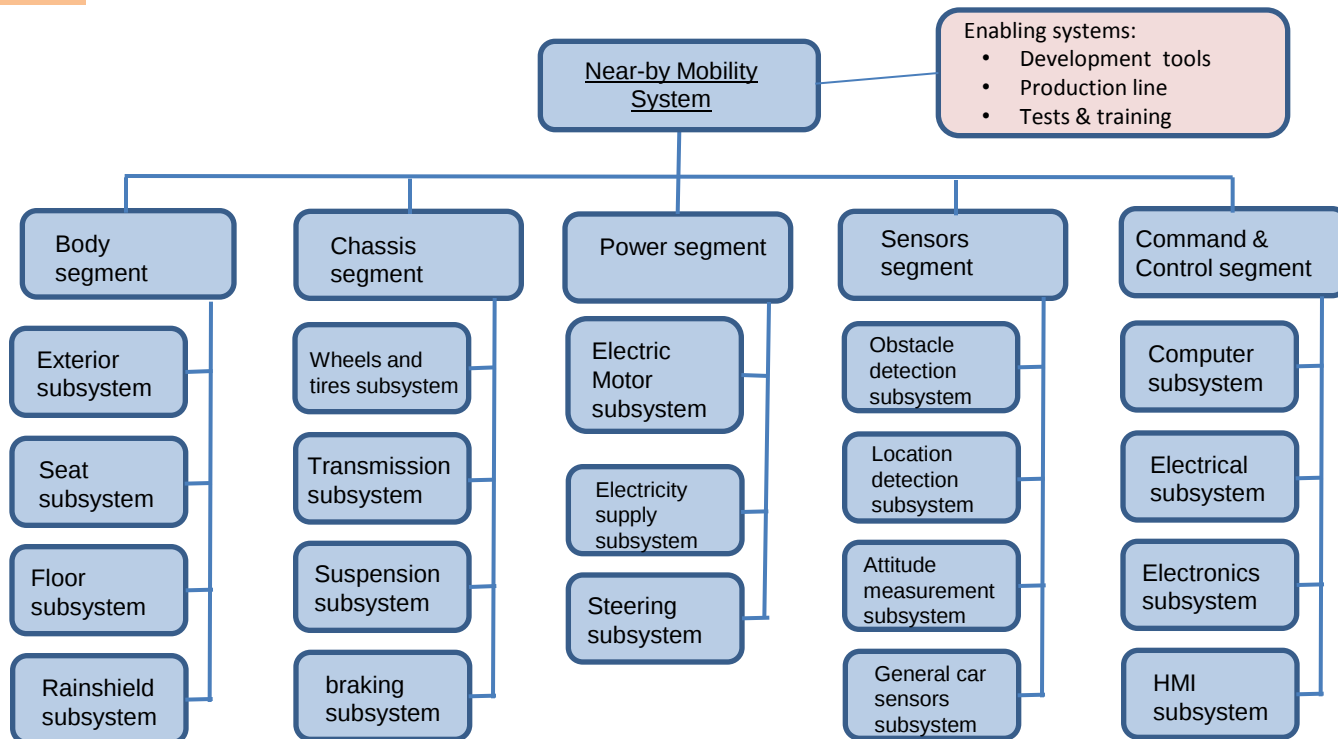
March 2011

Applying holisitic thinking / 1.21



Dr Joseph Kasser - National University of Singapore

החלטות תכן רוחביות ל- NbSMS



החלטות תכן רוחביות:

מערכת מודולארית/אינטגרלית:

- יחידת בקרה מרכזית + יחידות בקרה נפרדות למכלול ההנעה, מכלול החיישנים ולמכלול השלדה.
- מכלולי ההנעה, החיישנים והשלדה ידווחו BIT עצמי (התחלתי ומחזורי) ליחידת הבקרה המרכזית.
- חלוקת מטלות אדם/מכונה:

○ יכולת אוטונומית לניווט לנקודה על מפה – לאחר אישור מסלול ע"י הנוסע.

○ יכולת עצירה אוטומטית במקרה של מכשול.

○ יכולת בקרה של הנוסע על גודל וכיוון המהירות והתאוצה.

- הגדרת ממשקים: העברת המידע בין המכלולים תיעשה ב- message passing. המערכת תשתמש ב-

CANBUS בתצורת כוכב.

כל הזכויות שמורות לד"ר דרורה גושן ©

מחייב תוספת
קלט/פלט

ופונקציונאליות תואמת

פרק 2.3

הגדרת משימות, מאפיינים טכניים
וקונספט תפעול למכלולים

Mission statement:

NbSMS*

System main mission	Provide <u>affordable</u> self mobility capability at nearby neighborhood
System Secondary missions	Provide the mobility capability for diversified purposes
Key business goals	- Schedule – 12 month for 1st prototype - System target cost – 10,000 IS
Primary market	Market sector/group – elderly people
Secondary market	Currently or perceived – adults working from home, youth
Assumptions	Reasonable road conditions (no obstacles, stairs, sharp corners).
Risks	Competition
Issues	Definition of: diversified purposes and nearby neighborhood.

הגדרת משימות למכלולים

- יש להגדיר את המשימה הראשית של כל מכלול כך שאוסף המשימות יענה למשימה הראשית של המערכת:
- כנ"ל לגבי משימות המשנה
- יש להגיע להסכמה עם מהנדס המערכת של כל מכלול לגבי ההגדרה הכוללת של המשימה:
 - עלות היעד
 - לו"ז לאבי טיפוס למטרות שונות
 - שווקים רלבנטיים
 - הנחות
 - סיכונים
 - סוגיות

הגדרת המשימה למכלולים ראשיים

Mission statement:

NbSMS* Power Segment

System main mission	Provide rechargeable power capacity for the NbSMS
System Secondary missions	Provide steering capability for the NbSMS
Key business goals	- Schedule – 10 month for 1st prototype - System target cost – 3500 IS
Primary market	Market sector/group – small vehicles
Secondary market	Currently or perceived –
Assumptions	home charging capability
Risks	Competition
Issues	Definition of:

Mission statement:

NbSMS* Chassis Segment

System main mission	Provide smooth, stable and safe movement capability on paved and dirt roads.
System Secondary missions	Provide appropriate NbSMS dimensions. Provide transmission and suspension capabilities.
Key business goals	- Schedule – 10 month for 1st prototype - System target cost – 1500 IS
Primary market	Market sector/group – small cars
Secondary market	Currently or perceived –
Assumptions	Dirt and paved road conditions (no obstacles, stairs, sharp corners).
Risks	Competition
Issues	Definition of:

הגדרת מאפיינים טכניים

- יש להגדיר ולכמת את המאפיינים הטכניים העיקריים (*) של כל מכלול:
 - המאפיינים הנדרשים לביצוע המשימה (המשימות)
 - הביצועים הנדרשים לרמת המערכת
 - יש לבצע סימולציה/trade study
- יש להגיע להסכמה עם מהנדס המערכת של כל מכלול לגבי המאפיינים הטכניים

(*) – יש להקפיד שלא להגדיר דרישות תכן למכלול



Technical parameters definition

Technical parameters	description	Units/list	value	comments
dimensions (length, width, height)	External	meters	0.9X1.5X1.0	
vibrations	List of figures	Hz, Amplitude		See fig...
Passenger and payload weight	maximum	kg	200	
Angular inclination	maximum	degrees	30	
Automation & control (safety)	List	list	Table 1	
Maximum range & dynamic parameters	On paved road	Km & m/sec & m/sec ²	80	To be divided
Obstacle detection	any kind	m	1	
continuous operating time		hours	8	
styling and shaping (safety)	List	list	Table 2	
Maximum slope climb capability	On paved road	degrees	12	
reliability (MTBF)	Operating hours	hours	750	
purchase & operating cost		KIS+KIS/year	15+ 5	To be divided
Maneuvering capability on paved/dirt roads	List	list	Table 3	
Mean Time to repair (MTTR)		days	3	

Table 2
styling and shaping

Technical parameters	
Activation control	coded switch
Velocity control	Digital input
Input interface unit	Touch screen
Output interface unit	display
Warning indicators	Light & voice
Safety brakes operation	switch
Safety belts	yes
Air bags	yes
Right and left hand	yes
Rearward field of view	5m

Table 1

Automation & Control Requirements



Technical parameters	requirement
navigation	By target
Obstacle detection	Awareness indication & Automatic stop
Start-up activities	No more than 1
Driving activities	No more than 2
Stopping activities	No more than 1
Failure detection activities	No more than 4
Safety brakes activation	Specific events

Table 3
Maneuvering capability

Technical parameters	units	Paved/dirt roads (1)
Unpaved road (rock size)	mm	10
Single step size	cm	5
Puddle depth	cm	10
Rotation radius	cm	0



מאפיינים טכניים – NbSMS

מכלול השלדה Chassis Segment

- רדיוס סיבוב מינימאלי 0 (סיבוב במקום)
- עבירות בדרכים לא סלולות (בהתאם להגדרה)
- מבנה יציב עבור שיפועי צד של עד 30° ושיפוע קדמי/אחורי עד 40°
- מידות כלליות של $0.9 \times 1.5 \times 1.0$ מטר
- מרחק עצירה מקסימאלי של 1 מטר ממהירות של 15 קמ"ש במשקל של 500 ק"ג בירידה של 30°
- שיכוך רעידות הנובעות מתנאי הדרך לרמה הנדרשת בגרף הרצ"ב.
- מערכת תמסורת לשליטה במהירות הנסיעה. (אפשרות שליטה בשילוב המנוע)
- דיווחי מצב עצמי של המכלול כל דקה.

מכלול ההנעה Power Segment

- אנרגיה עצמית נדרשת להסעת 500 ק"ג במהירות של 15 קמ"ש למרחק אופקי של 80 ק"מ.
- הספק מרבי המתאים לנסיעה במהירות של 15 קמ"ש בשיפוע עלייה של 12°
- יכולת הטענה עצמית מלאה בבית (לאחר ריקון מלא) שתמשך 3 שעות.
- קצב זליגת אנרגיה באחסון ממושך עד 10% בשבוע.
- מערכת התנעה והדממה המקבלת פיקוד דיגיטלי חיצוני
- מערכת היגוי עצמאית המקבלת פיקוד דיגיטלי של מהירות ו/או תאוצה.
- דיווחי מצב עצמי של המכלול כל דקה.

הגדרת קונספט תפעולי כללי לכל מכלול

• הקונספט התפעולי הינו:

- הגדרה של כיצד המכלול יבצע את המשימה שלו במסגרת המשימה המערכתית
- הגדרת הסביבה התפעולית של המכלול
- הגדרת תפעול המכלול במהלך כל מחזור החיים שלו.

• עבור כל מכלול יש להגדיר:

- מגבלות תפעוליות
- סביבת ההפעלה
- גורמי ההפעלה
- קונספט התפעול
- יש להגיע להסכמה עם מהנדס המערכת של כל מכלול לגבי קונספט התפעול

(*) – יש להקפיד שלא להגדיר דרישות תכן למכלול



הגדרת סביבת ההפעלה

מכלול השלדה Chassis Segment

משימה: מכלול נע על גלגלים בצורה בטוחה, יציבה ונוחה, בעל יכולת עבירות, מידות מתאימות ויכולת היגוי.

- דרך סלולה/דרכי עפר/שבילי חצץ
- בית/מחסן
- מוסכים המטפלים בשלדה
- מכלול ההנעה
- מכלול החיישנים
- מכלול גוף הקלנועית
- מכלול הפקוד והבקרה

מכלול ההנעה Power Segment

משימה: מכלול בר-הטענה המספק אנרגיה חשמלית, הספק ותמסורת.

- מכלול השלדה
- מכלול הפקוד והבקרה
- מכלול גוף הקלנועית
- מכלול החיישנים
- תחנת הטענה חשמלית ביתית
- מוסכים המטפלים במנועים חשמליים

גורמי ההפעלה

גורמי ההפעלה במכלול ההנעה

- יחידת הפקוד והבקרה
- הטכנאי (תחזוקה)
- מערכת אספקת האנרגיה

גורמי ההפעלה במכלול השלדה

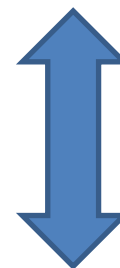
- יחידת הפקוד והבקרה
- הטכנאי (תחזוקה)

גורמי ההפעלה הינם גורמים חיצוניים
למערכת אשר מפעילים את המערכת:

○ מעבירים אותה בין מצבי פעולה
שונים.

○ מבצעים בה תחזוקה בשדה

גורמי הפעלה



תרחישי תפעול

ניידות עצמית
בסביבה קרובה



תפיסת ההפעלה



תפיסת ההפעלה הינה הגדרת **האופן בו המערכת פועלת** או מבצעת את משימותיה.

הגדרה זו מאגדת את **הנהלים והחוקים** לפיהם **גורמי ההפעלה** יפעילו את **המערכת בסביבתה** כך שהמשימה **תבוצע** לאורך כל **מחזור החיים** של המערכת.

- מכלול ההנעה יופעל ע"י מכלול הפיקוד והבקרה
- מכלול ההנעה יספק יכולת לאיתור תקלות ברמת LRU
- ניתן יהיה לפרק את מכלול ההנעה לצורך אחסון מהיר ע"י אדם בודד
- מכלול ההנעה יטען בבית בעזרת שקע חשמלי של כח (15 Amp מקסימום)
- גריסה תתבצע כמו במכונית חשמלית

- מכלול השלדה יופעל ע"י מכלול הפיקוד והבקרה
- מכלול השלדה יספק יכולת לאיתור תקלות ברמת LRU
- ניתן יהיה לפרק את מכלול השלדה לצורך אחסון מהיר ע"י אדם בודד
- גריסה תתבצע כמו במכונית רגילה

הגדרת "המכלול בסביבתו" (*)

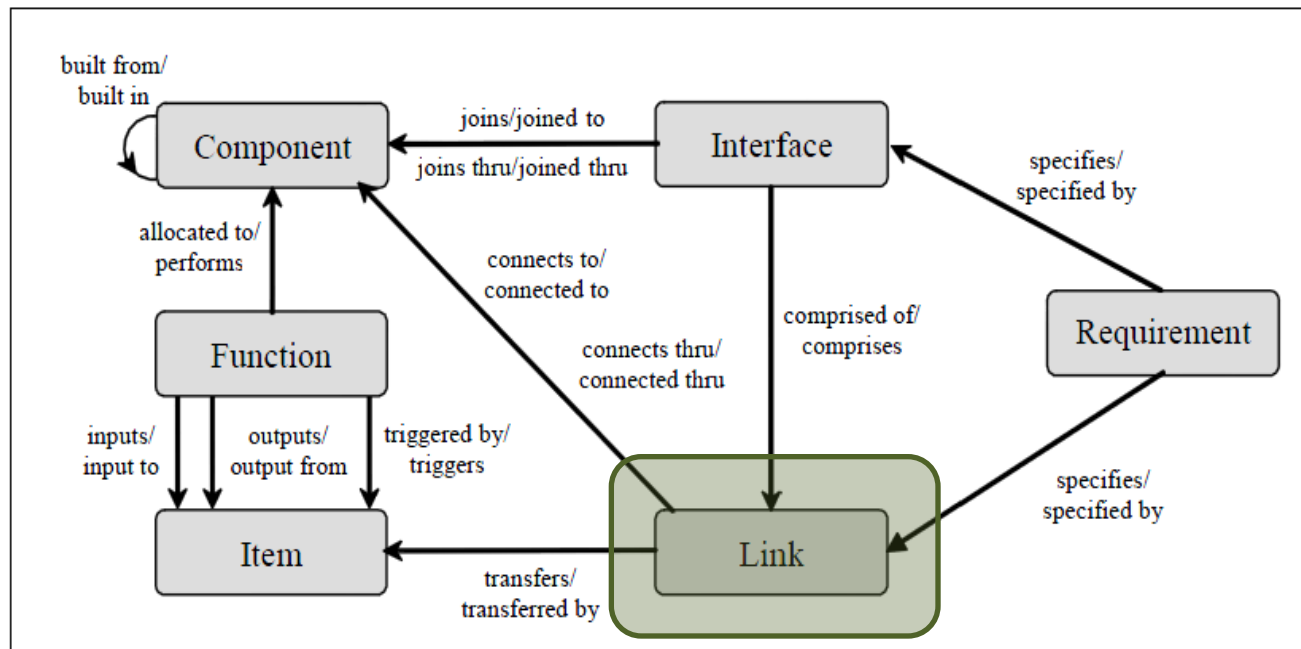
ממשקים בין מכלולים:
הגדרת חיבורים בין
מכלולים (קומפוננטות)
בעזרת הקשר link



החלטות התכנ המערכתיות
הקצאת המשימות
הגדרת המאפיינים הטכניים
הגדרת הקונספט התפעולי



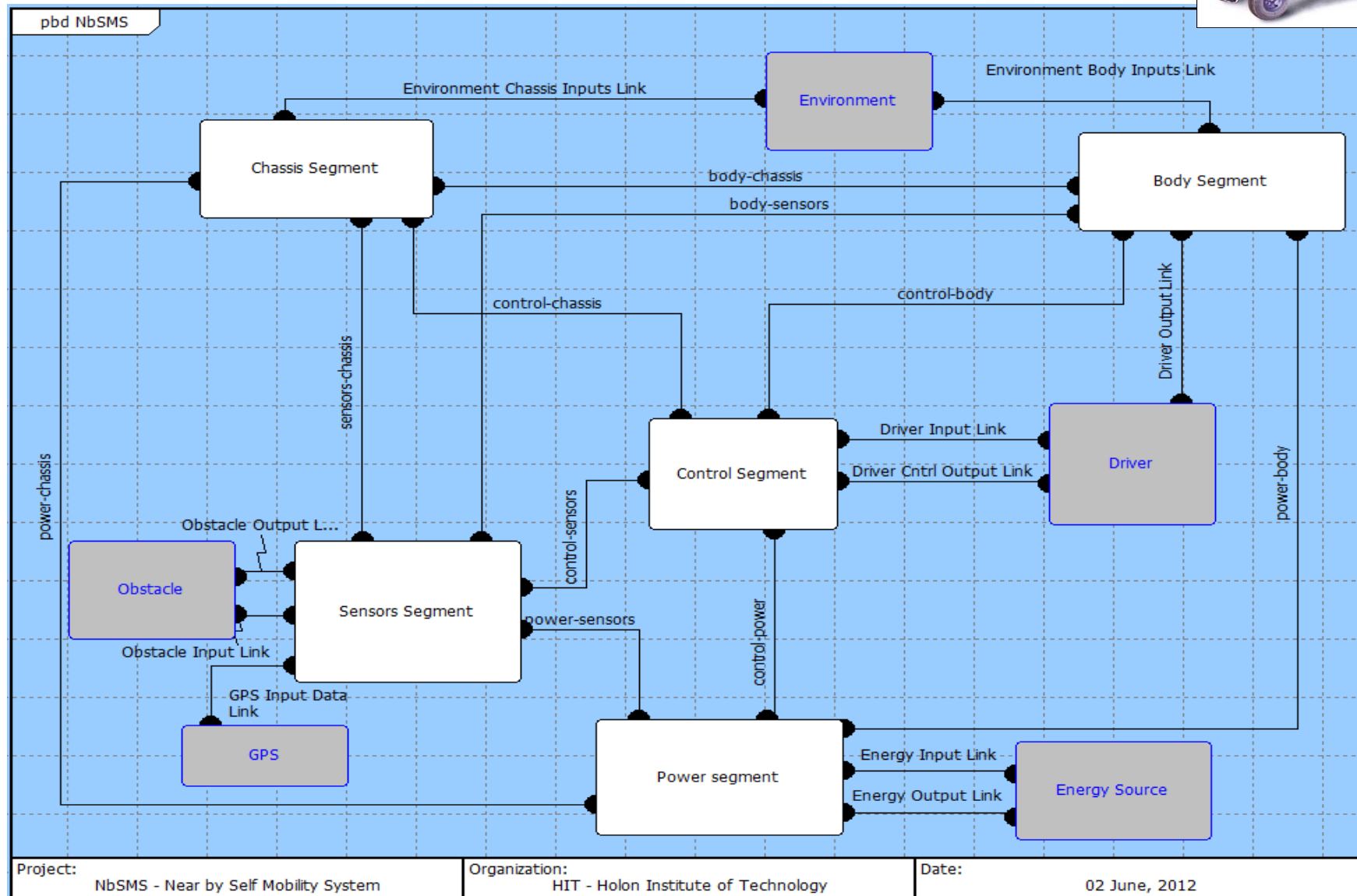
הגדרת מכלולים



(*) – המכלול כקופסא שחורה
כל הזכויות שמורות לד"ר דרורה גושן ©

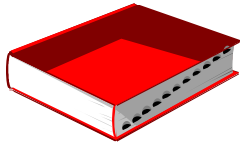


NbSMS Physical block diagram

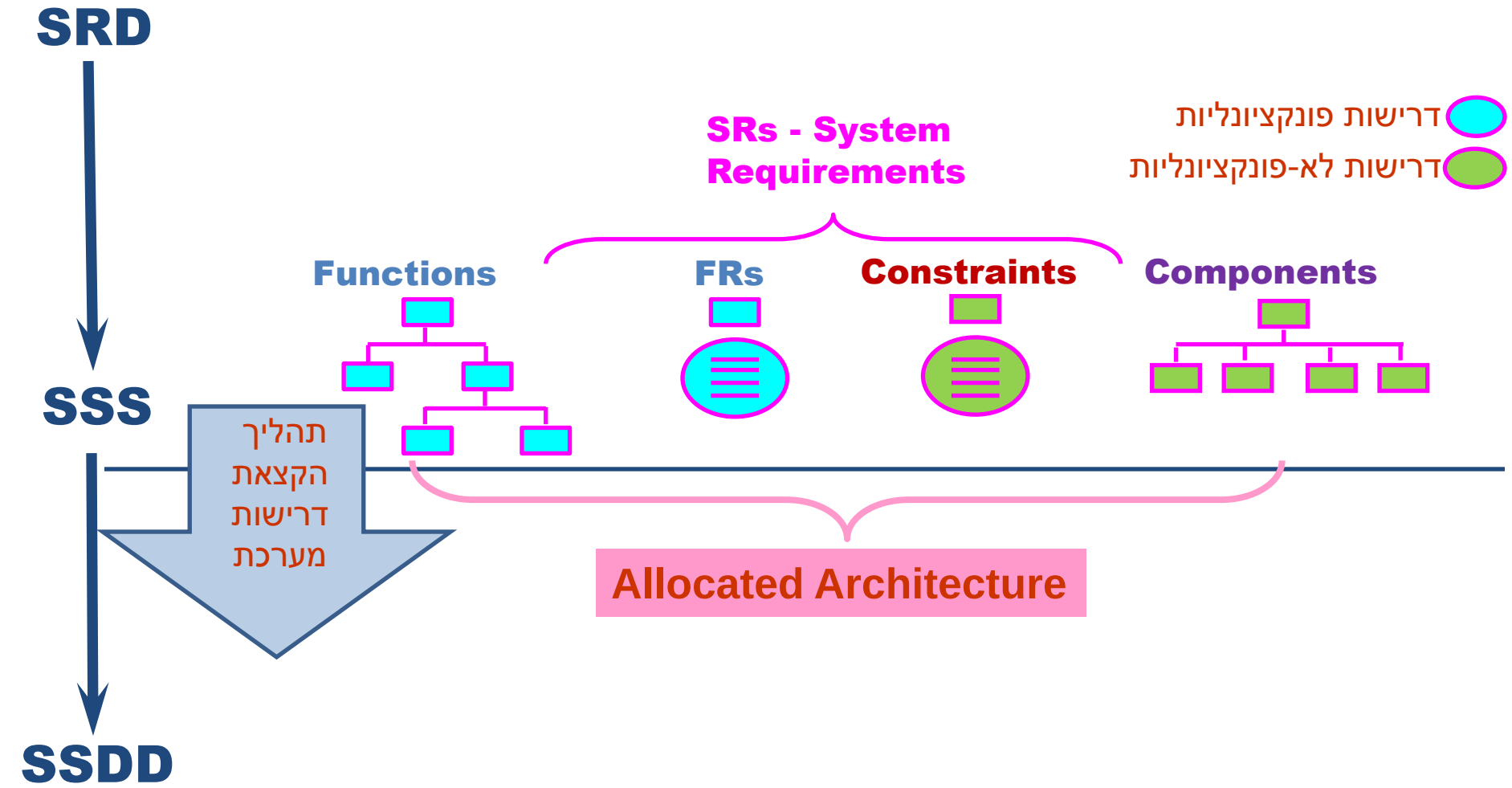


פרק 2.4

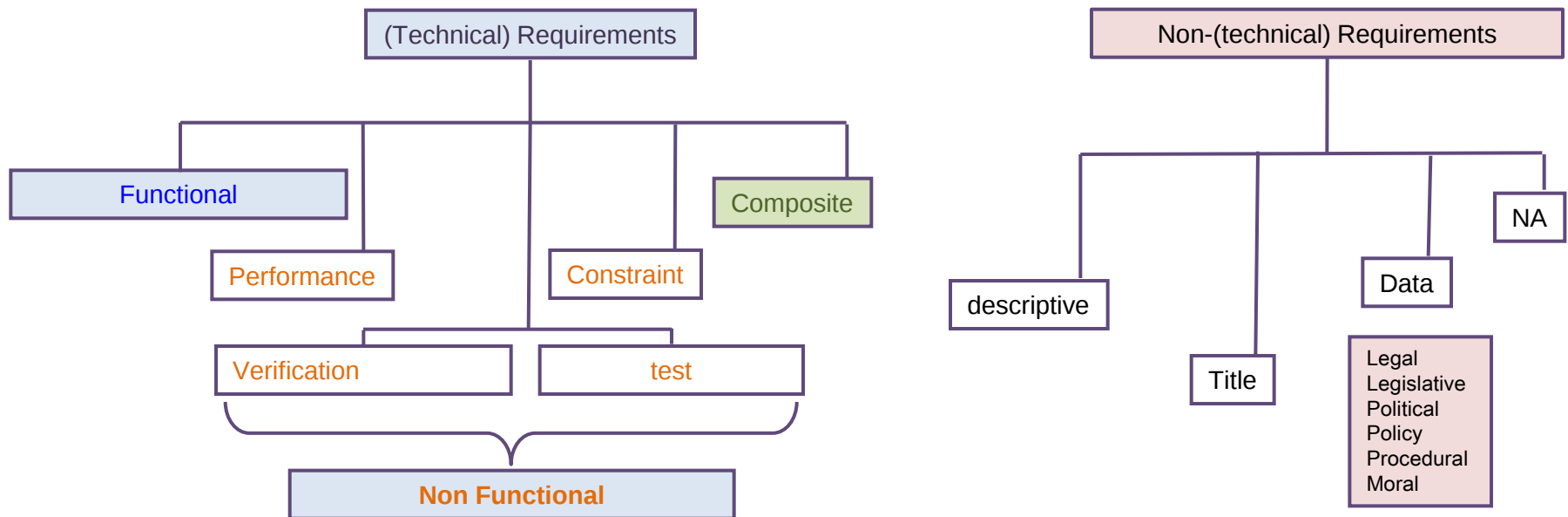
הקצאת דרישות לכל מכלול



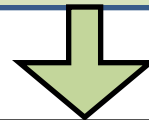
תהליך פתוח התכנ



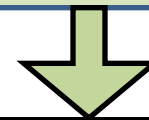
סיווג דרישות (Requirements type)



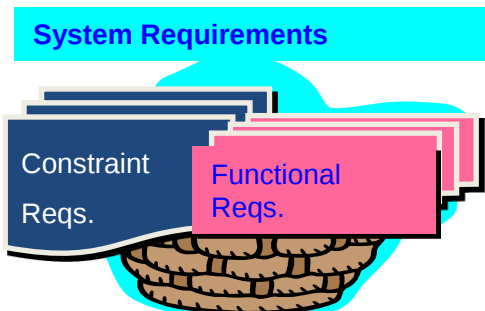
$$\text{SRs - System Requirements} = \text{FRs - Functional Requirements} + \text{Constraints}$$



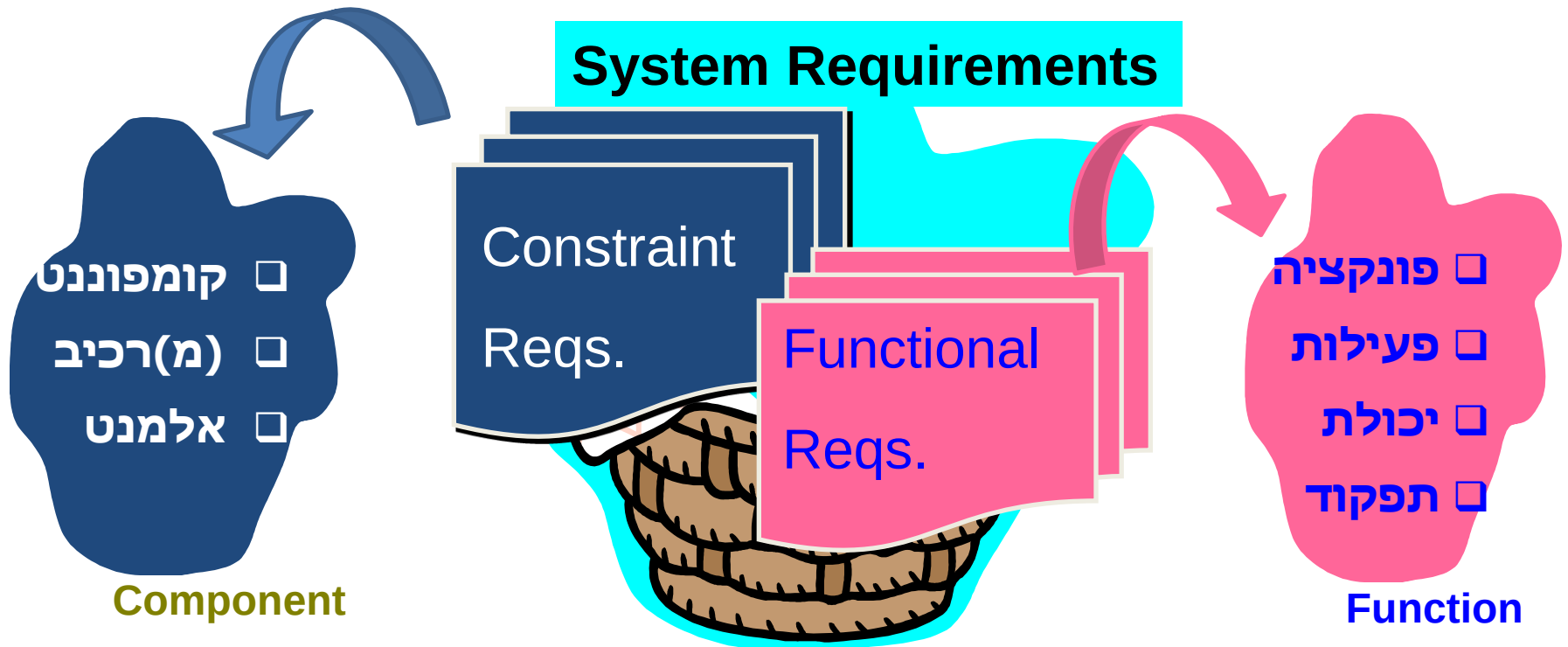
כל הדרישות הקשורות
לפונקציונאליות
(לתפקוד) של המערכת



כל הדרישות הקשורות
לביצועי המערכת,
לתצורה, לתכונות
הפיסיקליות וכל השאר



הגדרות נפוצות



A "component" is defined as a group of related requirements. The word "component" may be replaced with "element," "assembly," or other term useful for presenting the requirements. [MIL-STD-498]

A "capability" is defined as a group of related requirements. The word "capability" may be replaced with "function," "subject," "object," or other term useful for presenting the requirements. [MIL-STD-498]

הקצאת דרישות מערכת לא-פונקציונאליות

שיטות האימות	שיטות להקצאת דרישות
אנליזה טכנית-כמותית התומכת בבצוע החלוקה הספציפית	חלוקה (Apportionment) - דרישות המתחלקות בין תת-המערכות תוך שמירה על יחידות המידה שלהן: • עלויות • אמינות, זמינות • משקל, נפח
אין צורך באנליזה	שקילות (Equivalence) - דרישות מועברות לרמת תת-מערכת ללא שינוי: • צבע • שיטות עבודה
אנליזה פיזיקלית-טכנית התומכת בבצוע ההקצאה הספציפית. מלווה בדרך כלל בסימולציות מסוגים שונים.	מיזוג (Synthesis) - דרישות המתחלקות בין תת-המערכות ע"י שינוי ביחידות המידה שלהן: • דיוק • טווח • מהירות
מסתמכות על תהליך קבלת החלטות של מהנדס המערכת: • החלטות הנובעות משיקולי רכש • החלטות הנובעות משיקולים ארגוניים • הקצאות הנובעות מ"החלטות תכן מערכתיות"	הקצאת/הגדרת דרישות תכן חדשות הנוצרות בתהליך פתוח התכן (במיוחד כתוצאה מהחלטות התכן המערכתיות)

דוגמאות לחלוקת דרישות (שמירה על יחידות המידה)

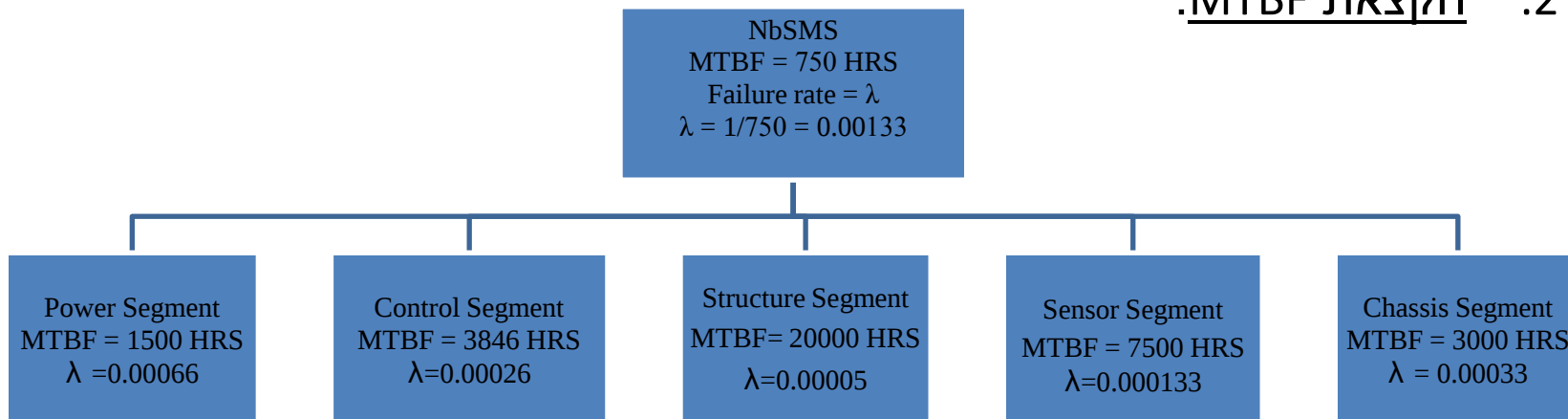
Requirements Allocation



1. הקצאת משקל: המשקל העצמי של הקלנועית הוא 300 ק"ג. יש לחלק משקל זה בין 5 תתי המערכות של הקלנועית:

- תת-מערכת ההנעה
- תת-מערכת השלדה
- תת-מערכת הבקרה
- תת-מערכת המבנה
- תת-מערכת החיישנים

2. הקצאת MTBF:



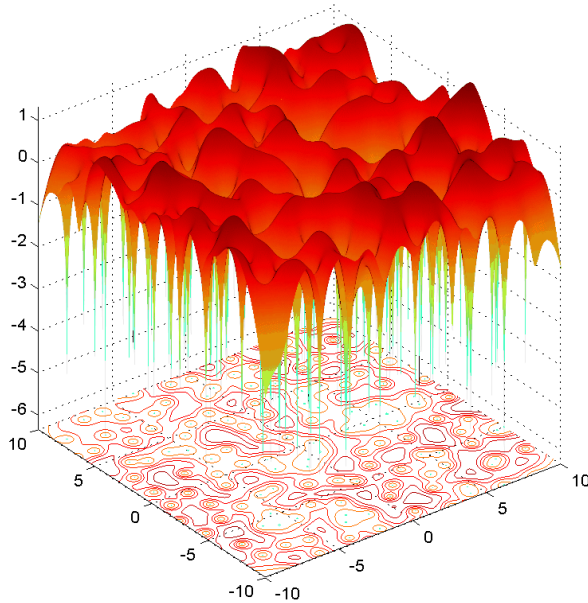
דוגמא למיוזוג דרישות (שינוי ביחידות המידה) Requirements Allocation



- הקצאת דיוק מיקום מכשול: במרחק של 1-50 מטר ב- CEP(*) של 10% וברזולוציה של 1 מ"מ. מרחק גילוי מקסימלי – 50 מטר.
- הדרישה תתחלק בצורה הבאה:
 - חיישן מרחק – דיוק של 0.07 מטר
 - חיישן כיוון – דיוק של 0.07 רדיאן
 - אורך מילה בתקשורת: 1 מ"מ ועד ל- 50 מטר == 16 סיביות.
 - יכולת תאוצה קבועה של 0.1 m/sec^2 :
 - מרחק עצירה ממהירות של 15 קמ"ש - 0.8 מטר תוך 4 שניות.
 - מהירות תקשורת ועיבוד נתונים – 0.1 שניות.

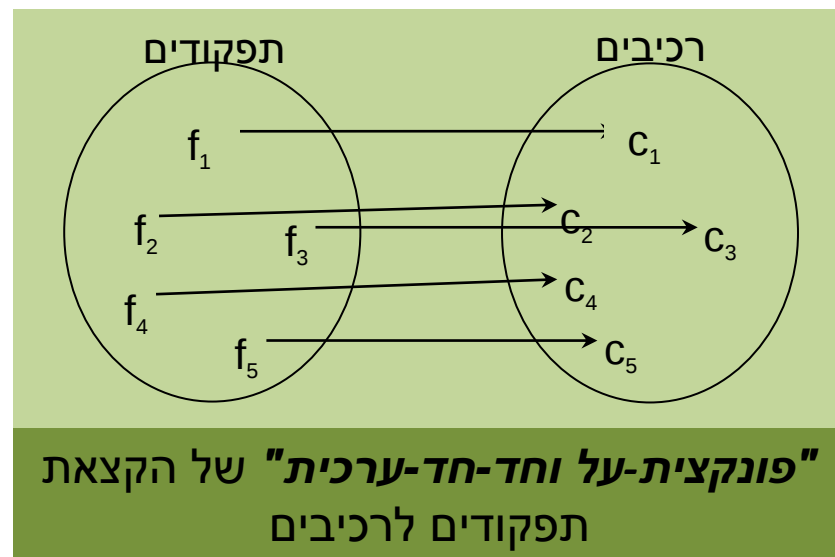
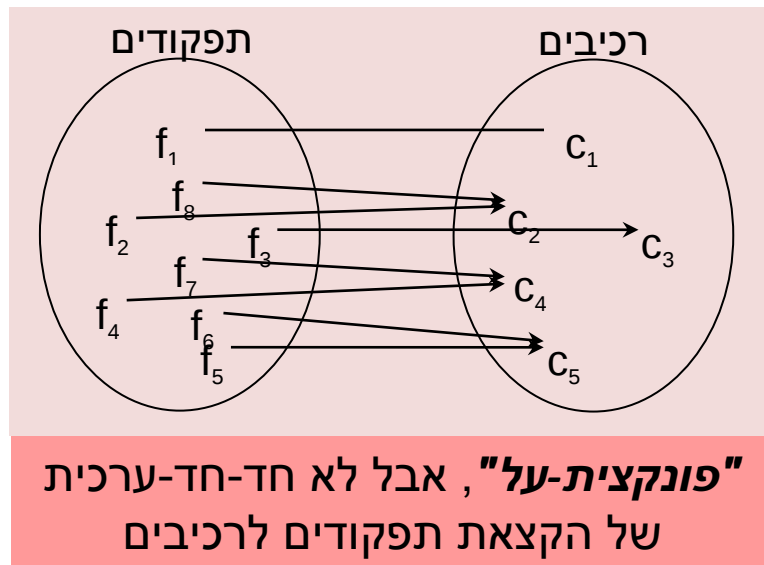
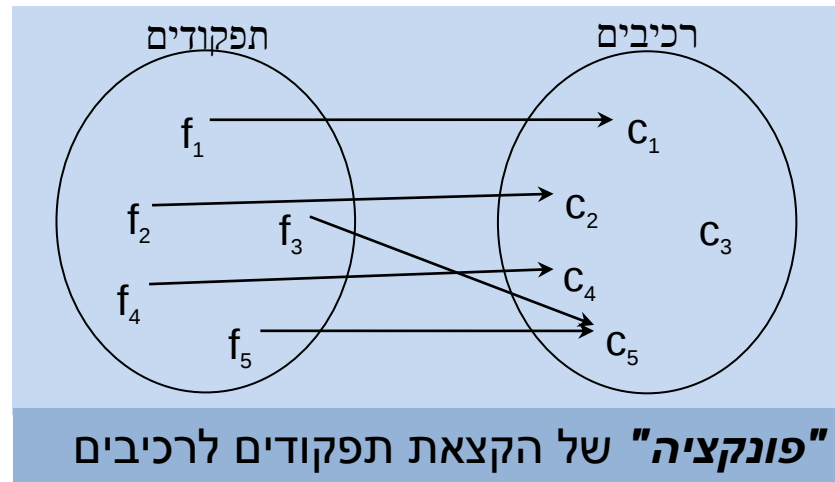
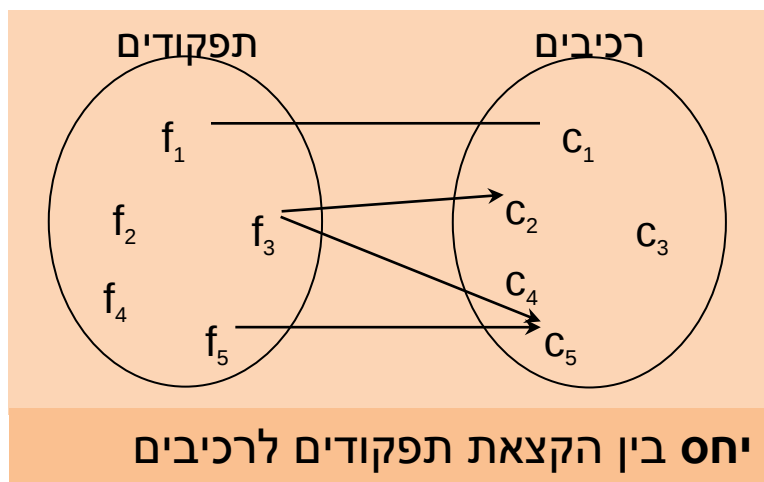
(*) CEP – Circular Error Probability

הקצאת דרישות מערכת פונקציונליות

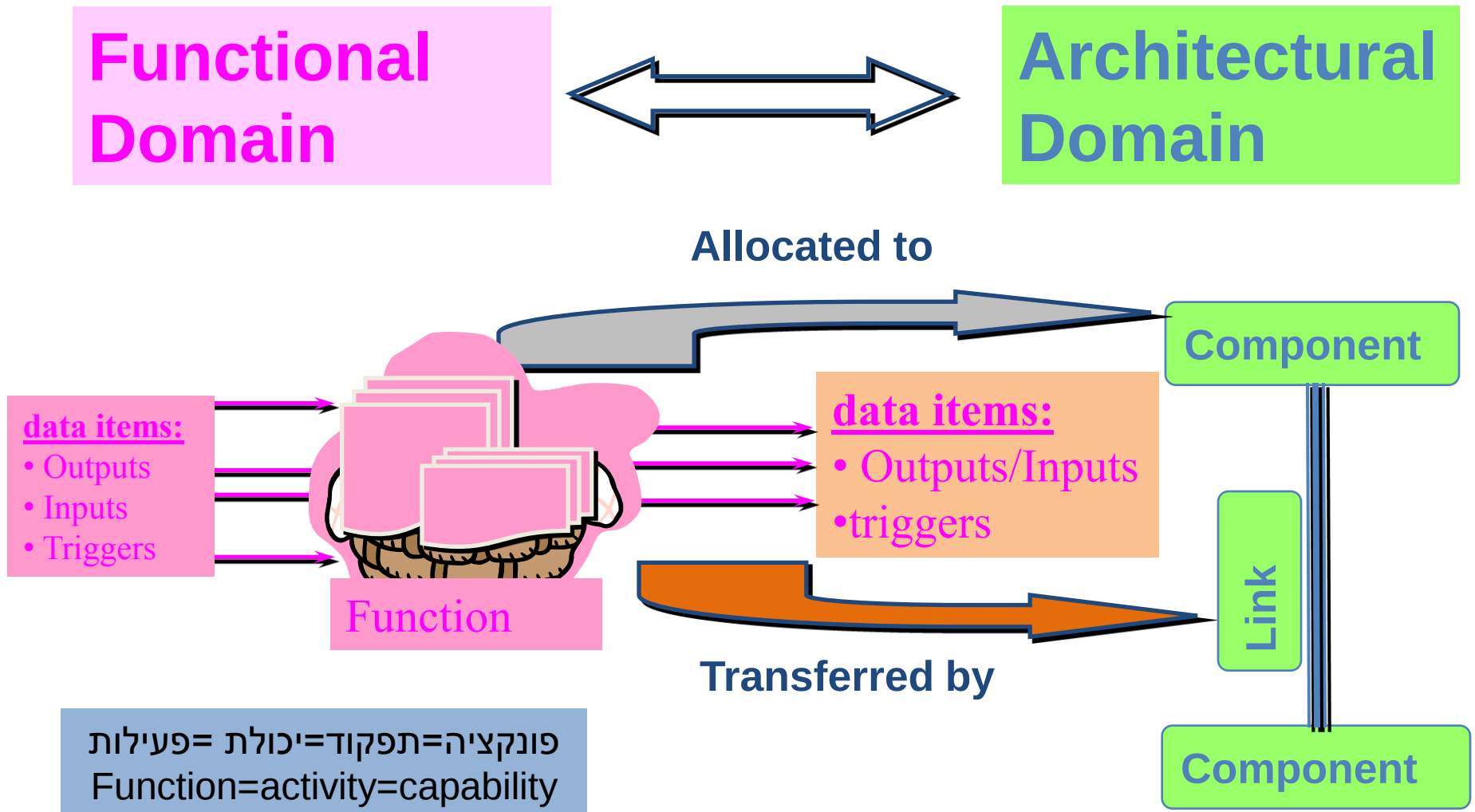


- תהליך הקצאת הדרישות הפונקציונליות הינו תהליך הכולל 3 שלבים:
- שלב א – חלוקת המודל הפונקציונלי (פונקציות העלה) למכלולים הפיסיקליים/קונספטואליים.
- שלב ב – הגדרת חיבורים (links) בין המכלולים והקצאת קלטים/פלטטים לכל חיבור בהתאם להקצאת הפונקציות.
- שלב ג – הקצאת הדרישות הפונקציונליות לפונקציות העל של כל אחד מהמכלולים בהתאם להקצאת הפונקציות.

הגדרות מתמטיות ל"פונקציות" ההקצאה



תהליך הקצאת הפונקציות/התפקודים



הקצאת התפקודים למכלולים

- הקצאת התפקודים מתבצעת לאחר:
 - הגדרת המשימה, המאפיינים הטכניים וקונספט התפעול של כל מכלול
 - הגדרת "המכלול בסביבתו"
- העקרונות המנחים (כללי "אצבע") לביצוע הקצאת הפונקציות:

– המטרה הינה לשייך כל פונקצית-עלה לקומפוננטה אחת בלבד. אם יש צורך בשיוך לקומפוננטת נוספות – יש להמשיך ולחלק את פונקצית העלה לפונקציות עלה נוספות.

– מינימום אינטראקציה (צימוד) בין המכלולים השונים:

• מינימום ממשקים בין מכלולים – שימוש בדיאגרמות N^2 לבדיקת צימודי ממשק

– מניעת בעיות קריטיות הקשורות לתפעול ולבקרת הפונקציות ע"י סנכרון ותזמון נכון של תהליכי הבקרה

– רובוסטיות לשינויים פיסיקליים (כל עוד נשמרת הפונקציונאליות):

• זיהוי פונקציות/קומפוננטות הנוטות לשינויים

• זיהוי פונקציות/קומפוננטות העומדות בפני שינוי טכנולוגי

– "הגיון" פנימי אחיד לביצוע ההקצאה.

– התאמה למבנה ארגוני (קבלני משנה וכו').

יתכן שידרשו
כמה איטרציות
לקבלת הקצאה
אופטימלית

הקצאת הפונקציות - NbSMS



Name	performs
Chassis Segment	F.1.1.1 Detect Vibration F.1.1.2 Dampen Vibration F.1.7.1.1 Change Wheel Direction F.1.7.1.2 Emergency Change of Wheel Drection F.1.7.2.3 Stop Structure F.1.7.3.4 Move Structure perform chassis segment function <i>Behavior Type: Integrated (Root)</i>
Body Segment	F.1.4.1 Perform Body Function perform body segment function <i>Behavior Type: Integrated (Root)</i>
Control Segment	F.1.2.1 Calibrate System & Perform INIT BIT F.1.2.2 Perform Periodic BIT F.1.2.3 Perform Maintanance Routines F.1.3 Provide Training Capability F.1.5.1 Accept, process & acknowledge Driver Data and Cmnds F.1.5.2 Process & Display Info F.1.6.1.1 Calculate Trajectory to Required Waypoint F.1.6.1.2 Calculate Alternative Trajectories perform control segment function <i>Behavior Type: Integrated (Root)</i>

Power segment	F.1.7.2.1 Calculate & provide Emergency Stop Thrust F.1.7.2.2 Calculate & provide Required Stop Thrust F.1.7.3.3 calculate & provide directional Thrust F.1.7.4.1 Accept & Store Energy F.1.7.4.2 Convert Energy to Motion F.1.7.4.3 Handle, Calculate & Indicate Energy Status F.1.7.4.4 Handle Battery perform power segment function <i>Behavior Type: Integrated (Root)</i>
Sensors Segment	F.1.4.2 Perform Different Sensors Function F.1.6.2.1 Calculate Obstacle Location F.1.6.2.2 Provide Detection Signal F.1.7.3.1 Calculate Self Location F.1.7.3.2 Calculate Self State Vector perform sensors segment function <i>Behavior Type: Integrated (Root)</i>

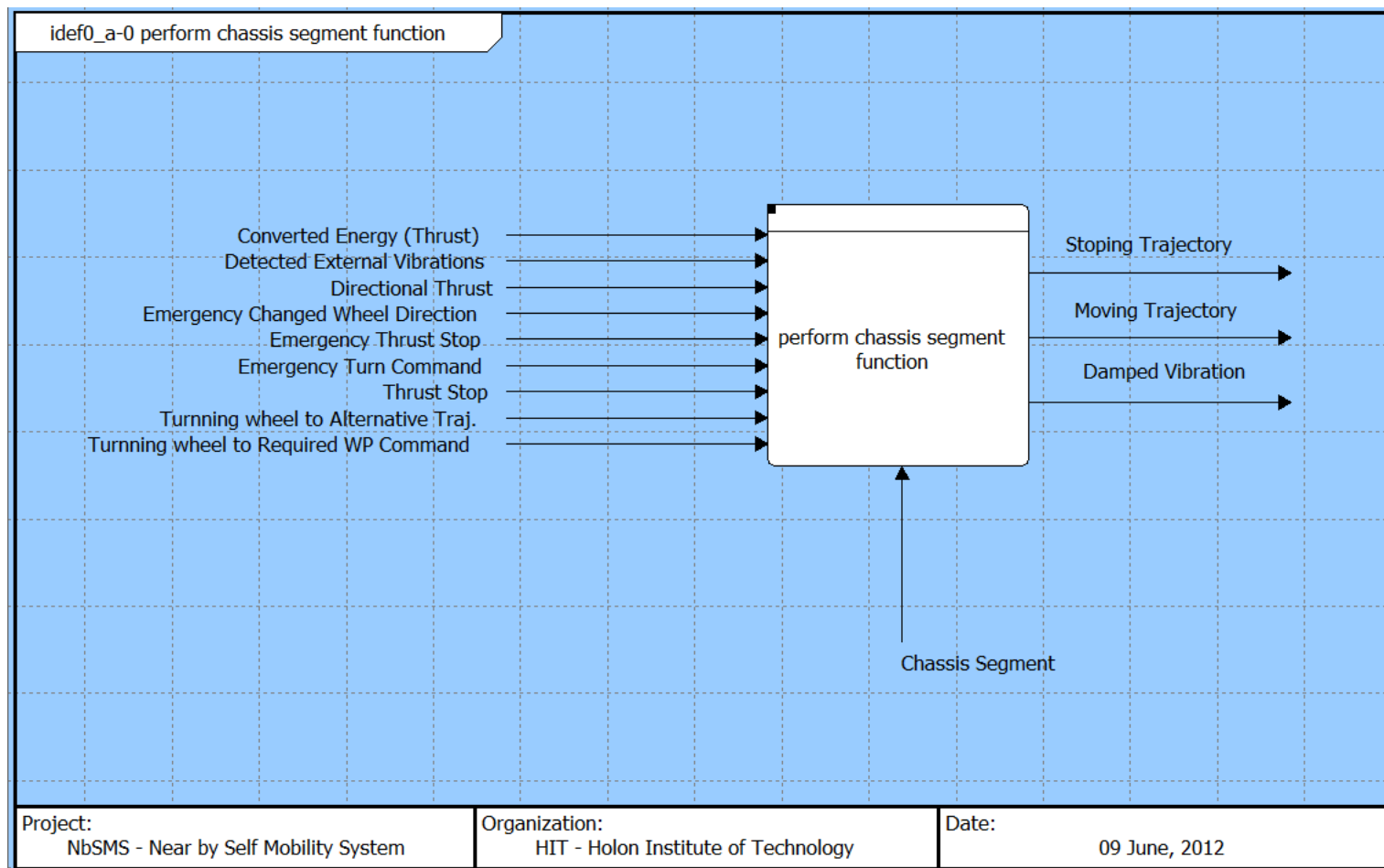
Sort: Numeric

חיבור המכלול לסביבה

- יש לשרטט כל מכלול כ"קופסא שחורה" בתוך הסביבה התפעולית שלו
- יש להגדיר את זרימת המידע אל ומתוך ה"קופסא השחורה" למערכות הפיסיקליות סביבו.
- לקבלת הממשק הפונקציונאלי:
 - מעתיקים את הפונקציות המוקצות למכלול (הבסיסיות) לתוך ה"קופסא השחורה".
 - משרטטים את המכלול הפונקציונאלי בסביבתו:
 - מחברים את הכניסות מהמערכות החיצוניות לפונקציות המתאימות
 - מחברים את היציאות מהפונקציות המתאימות אל המערכות החיצוניות
 - מוסיפים את הכניסות והיציאות הראשיות בין הפונקציות
- בבסיס מידע ממוחשב: מבצעים הנ"ל בעזרת EFFBD ודיאגרמות N^2



המכלול בסביבתו

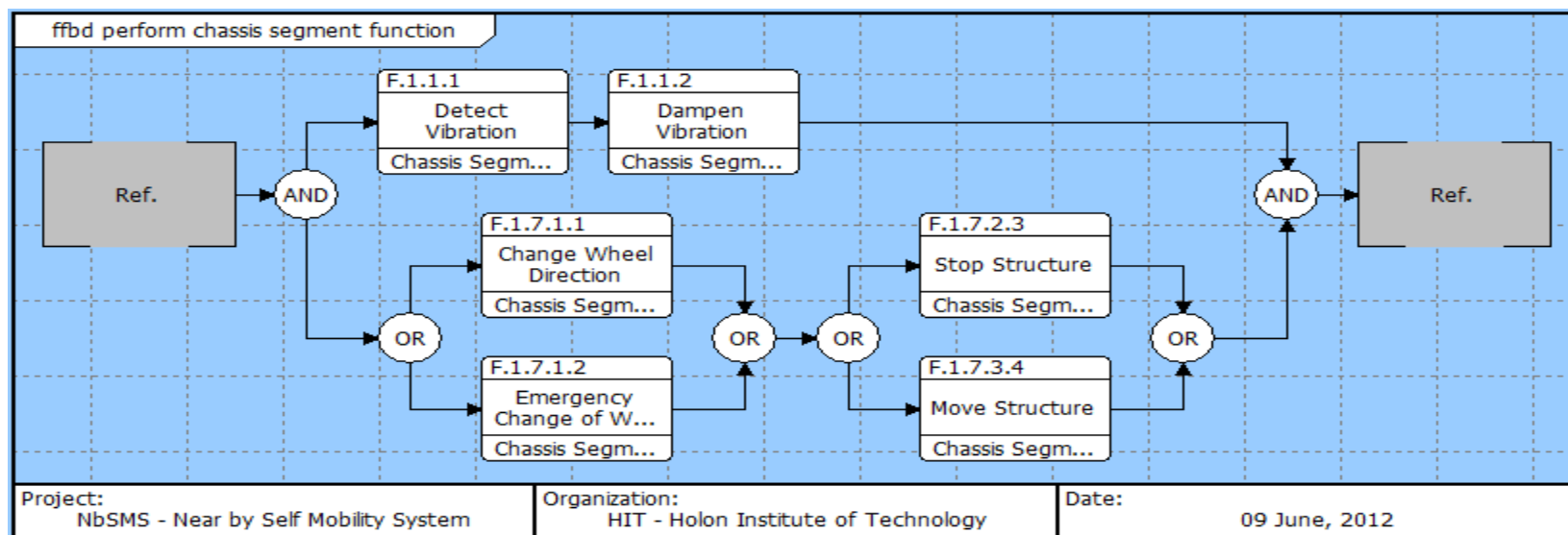
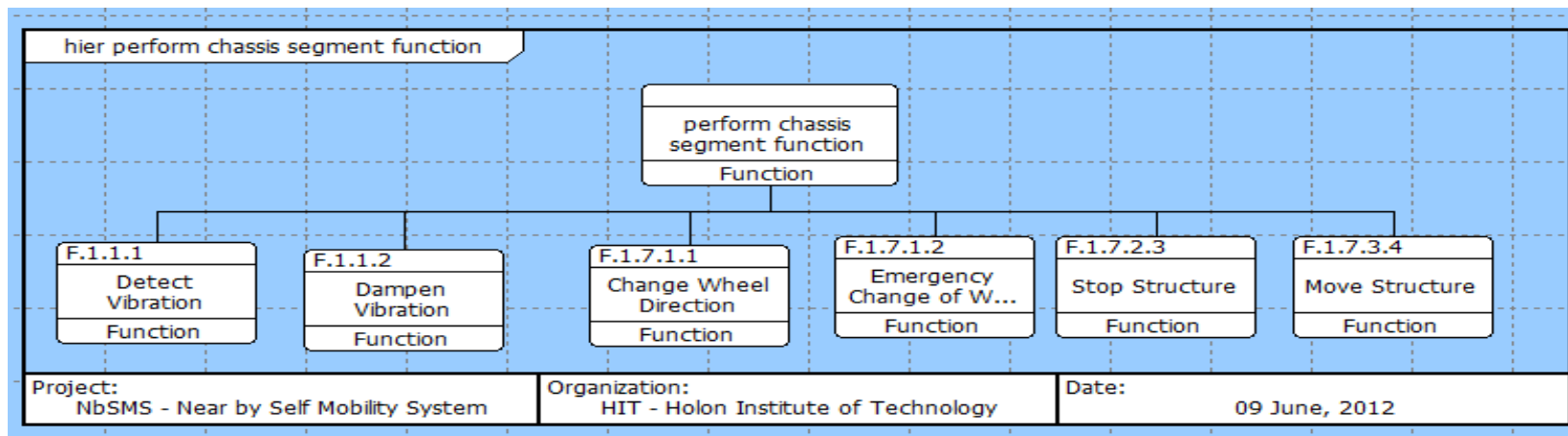


הגדרת תרחישים לכל מכלול

- הגדרת התרחישים מתבצעת תוך שימוש בפונקציות העלה המוקצות לכל מכלול
- יש להגדיר תרחישים עיקריים בלבד הנצפים ברמת המערכת (חיצוניים)
- יש להוסיף פונקציות נדרשות לצורך התרחישים ולעדכן את המודל הפונקציונאלי בהתאם
- יש לעדכן את תרשימי הזרימה בין המכלולים בהתאם לתרחישים

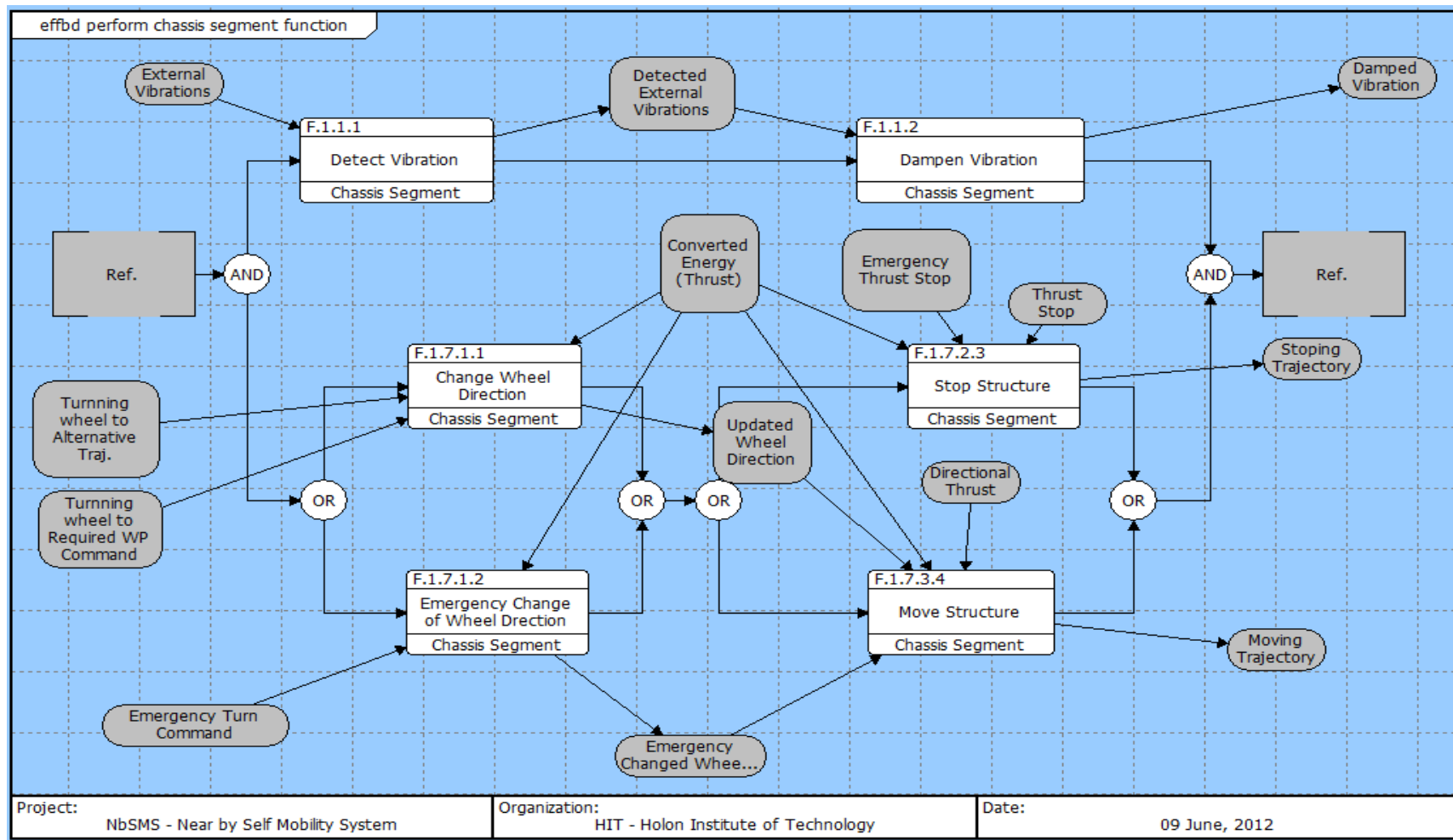


Chassis segment function





Perform chassis segment funct



פרק 3

השלמת בסיס המידע ההנדסי

השלמת בסיס המידע

תהליך הקצאת "דרישות המערכת" (SRs) למכלולים (*)

- דרישות פונקציונאליות קיימות:

- אם הפונקציה עוברת בשלמותה למכלול - כל ה-FRs שלה מוקצים לאותו מכלול (שינויי שם בלבד)
- אם הפונקציה מתחלקת בין כמה מכלולים - יש לחלק אותה **במודל המערכתי** לתתי פונקציות עם שיכתוב מתאים.

- דרישות לא-פונקציונאליות (Constraints) קיימות:

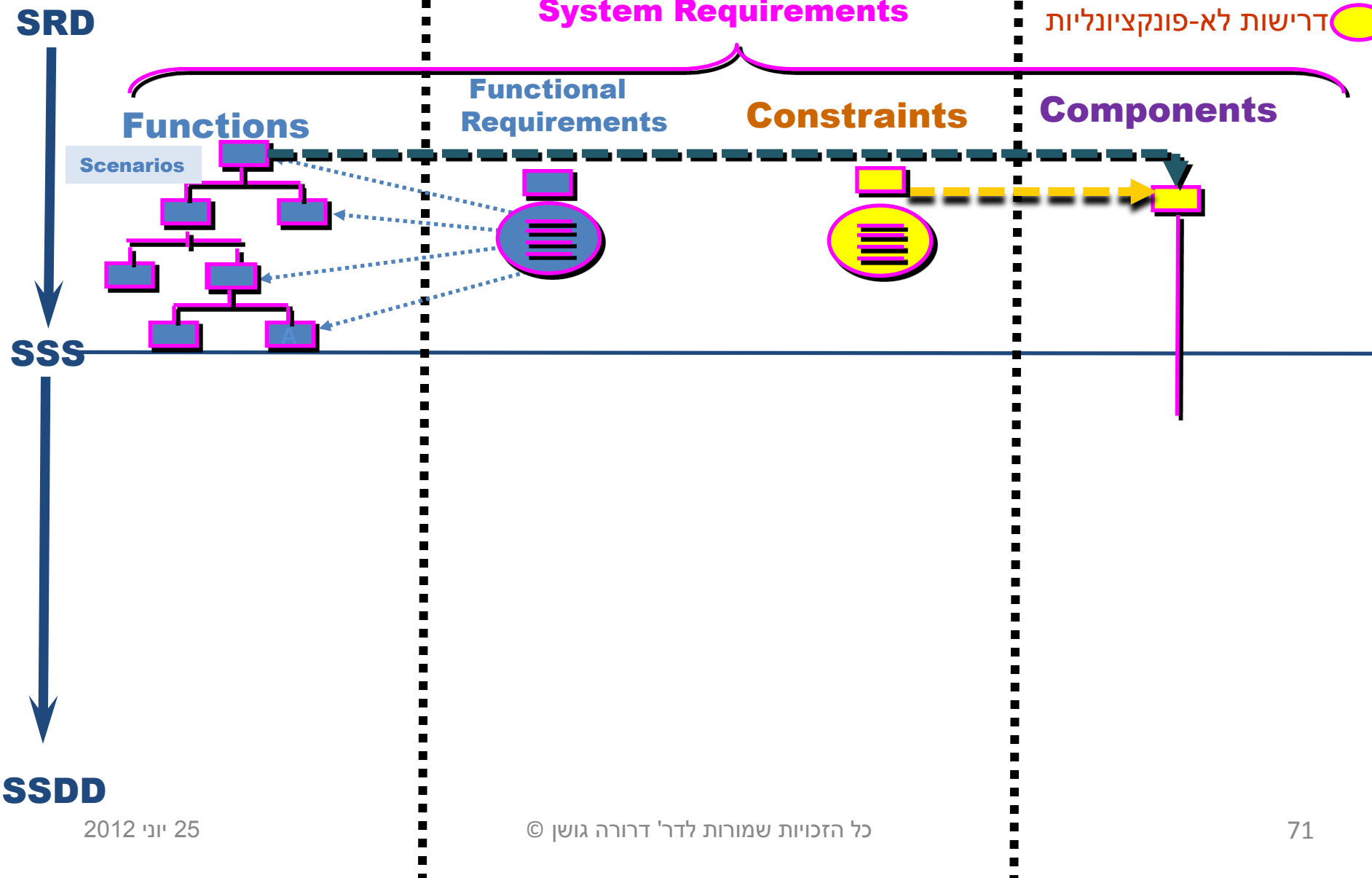
- ישנן דרישות העוברות ישירות למכלולים הרלבנטיים
- ישנן דרישות עליהן יש לבצע אנליזות מסוגים שונים כדי ל"חלק" אותם בין המכלולים הרלבנטיים

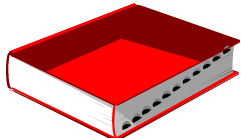
- דרישות נגזרות

- בתהליך הקצאת הפונקציות ובניית התרחישים נוצרות דרישות פונקציונאליות ולא-פונקציונאליות חדשות. יש לכתוב דרישות אלו בצורה נכונה בבסיס הנתונים.

בסיום תהליך פתוח הדרישות

דרישות פונקציונליות





מבנה בסיס המידע

בסיס מידע לתהליך פיתוח המערכת

דרישות פונקציונליות
דרישות לא-פונקציונליות

SRD

System Requirements

SSS

SSDD

Functions

Functional Requirements

Constraints

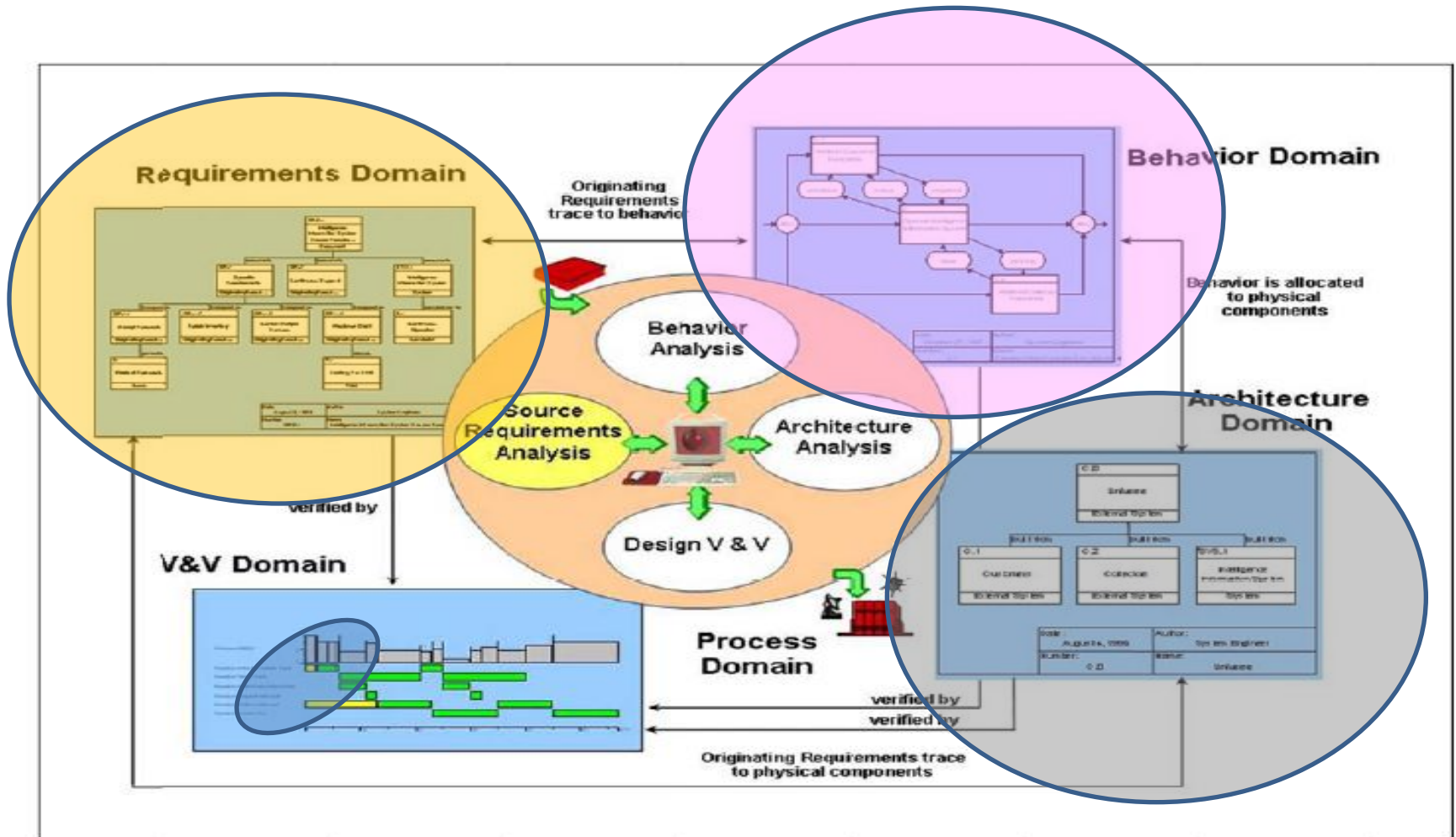
Components

Scenarios

System Wide Design Decisions

CSCI
CSCI/LRU

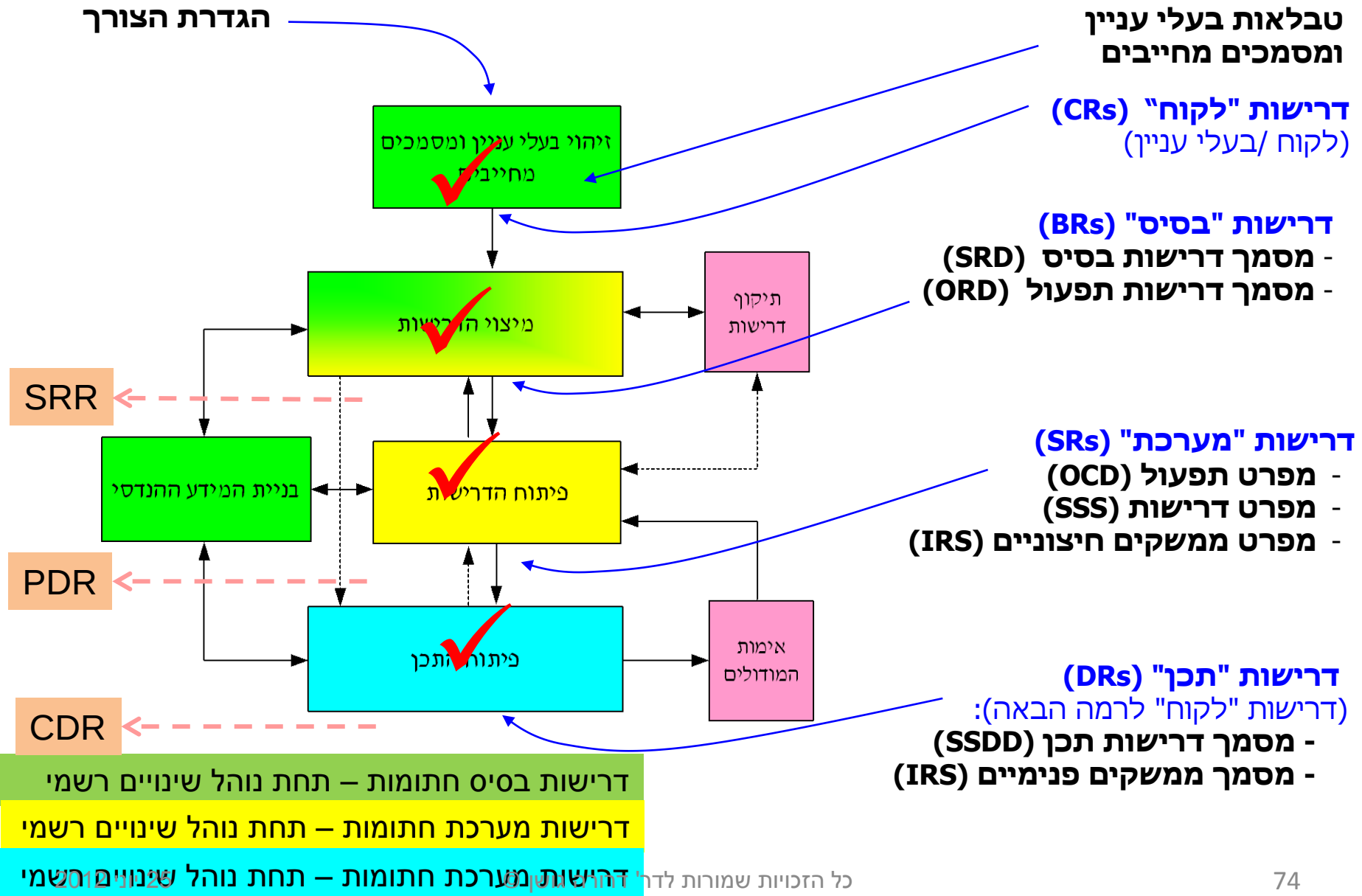
מבנה בסיס המידע ההנדסי ב- CORE



MBSE Activities

תהליך התכנון הנדסי מבוצע תוך בניית המודל המערכתי: מודל המבוסס על מבנה מקושר של דרישות, פונקציות, רכיבים ובדיקות. המודל המלא כולל גם תרחישים מערכתיים, ממשקי חומרה ותוכנה וכמובן פלטים/קלטים.

תהליך התכן ההנדסי ותוצריו



פרק 4

אימות ותיקוף התכן ההנדסי

פיתוח התכנה הנדסי



SRD = System Requirements Document
SSS = System/Segment Specification
SSDD = System/Segment Design Document
SRS = software Requirement specification
LRU Spec = Line Replaceable Unit Specification

בדיקת בסיס הנתונים

הרצת תסריטי
תוכנה - scripts

- בדיקות "ניקיון" בסיס הנתונים:
 - בדיקת קישוריות של דרישות התכן של כל מכלול
 - בדיקת עקיבות של זרימת המידע בין המכלולים
 - בדיקת קיום דרישות אימות לכל דרישת תכן
- בדיקות סטטיסטיות של בסיס המידע

תסריטי תוכנה (Scripts)

- מקפלים לתוכם את כל "רזי המתודולוגיה".
- נקודת המוצא: הכלי **אינו חוסם** ב- OnLine ביצוע קישורים הנוגדים את המתודולוגיה או שימוש במידע חלקי.
- הפעולה הנדרשת: בכל הרצת תסריט תוכנה מתבצעות בדיקות של נכונות קישורי המידע ב- DB = בדיקות של **איכות המידע**.
- בדיקות אפשריות:
 - **חוסר התאמה** של המידע שהוכנס ביחס למתודולוגיה - שגיאות בקשרים שהוקמו,
 - **חוסר בקשרים** נדרשים וכן מידע אחר שחסר לשם **הפקת מסמך**.
 - **חיווי לגבי חוסר של כיסוי** לדרישות החוזה
 - **קישורים שגויים** בין דרישות ברמות פיתוח שונות
 - **קשרים** בין אלמנטים למסמך בו הם מופיעים
 - מעקב "כשרות" (**קיום ECRD מאושר**) של תוספת/הסרה של אלמנטים ביחס להקפאות
 - דו"חות **סטאטוס** תקופתיים



בדיקות קישוריות של הדרישות והקצאות של התפקודים

Unaddressed leaf-level requirements Query

Lists any leaf-level requirements that are neither refined by requirement, specify something, nor the basis of anything

- תסריט התוכנה (Script) בודק שכל דרישה נובעת מדרישה קודמת או מגדירה/מהווה בסיס לאלמנט תכנוני (פונקציה, מידע, ממשק או קומפוננטה) או לדרישת אימות:
- כל ה- CRs קשורים דו-כיוונית ל- BRs
- כל ה- BRs קשורים דו-כיוונית/הופכים להיות SRs (כלומר, ORs, FRs או Constraints):
- כל הדרישות החדשות (derived requirements) הן FRs (הקשורות לפונקציות המתאימות) או Constraints (הקשורים למרכיב העל).

Unaddressed leaf-level function Query

Lists in the transcript all leaf-level functions that are not allocated to a component

- תסריט התוכנה (Script) בודק שכל תפקוד - עלה מוקצה למרכיב (קומפוננטה)

בדיקת קיום דרישות אימות

Unverified leaf-level requirements Query

Lists any leaf-level requirements that are not verified by verification requirements

– תסריט התוכנה (Script) בודק שלכל דרישת-עלה מוגדרת שיטת אימות

בדיקת עקיבות זרימת המידע

Item Consistency Check v1.0

checks the consistency of I/O between a function and its sub-functions.

– תסריט התוכנה (Script) בודק את זרימת המידע (items) המוגדרת בין הפונקציות.

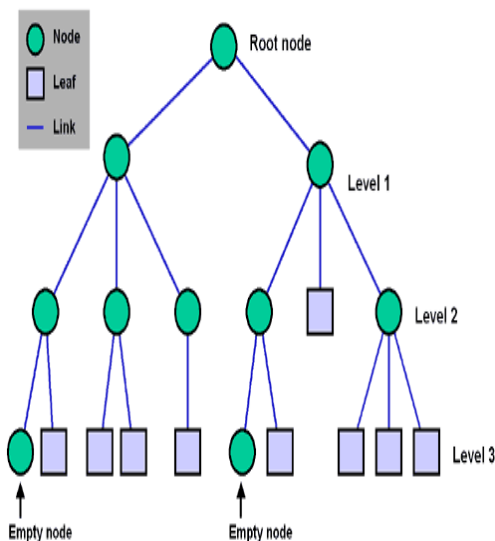
– הבדיקה מאשרת/מתריעה אם המידע קשור דו-כיוונית:

- כל הפונקציות החיצוניות מחוברות לפונקציות המערכתיות הרלבנטיות.
- כל הקלטים/פלטים מהפונקציות החיצוניות מחוברות לפונקציות המערכת הרלבנטיים
- כל הקלטים/פלטים בין הפונקציות המערכתיות הראשיות מוגדרים וקשורים
- כל הקלטים/פלטים בין הפונקציות הפנימיות מוגדרים וקשורים

בדיקת רקורסיביות (*)

Recursive Element Check v1.0

For user selected element class, checks for any recursion in any hierarchies formed by the parent-child relationship. Results are written to the Transcript.



• תסריט התוכנה בודק אם קיימים מסלולים מעגליים או לולאות עצמיות בכל אחד מהעצים המכוונים שנבנו בבסיס הנתונים:

- בעץ הדרישות
- בעץ הפונקציות
- בעץ המרכיבים
- בעץ פריטי הזרימה

(*) – לולאות חוזרות בעצים המוגדרים לאלמנטים השונים

סטטיסטיקה של בסיס הנתונים

Database Statistics Report v1.1

Outputs scheme utilization information for a user selected facility

Class	Number
Component	52
Concern	11
Document	12
Function	69
Interface	10
Item	64
Link	21
Organization	14
ProgramElement	11
Requirement	195
Resource	1
Risk	4
State/Mode	16
VerificationEvent	8
VerificationRequirement	76

- יש לבדוק ולעקוב ברציפות אחרי הסטאטוס (או הסטטיסטיקה של) בסיס הנתונים.
- השיטה: הרצת תסריטי תוכנה (Scripts) שונים.
- בגרסת האוניברסיטה של CORE קיים תסריט אחד בלבד לסטטיסטיקה.
- הנתונים העיקריים הנדרשים:
 - מספר הדרישות הכולל
 - מספר הפונקציות המוגדרות
 - מספר המרכיבים (Components)
 - מספר הממשקים החיצוניים (links, external items, external issues)
 - מספר הסוגיות הפתוחות (issues)
 - מספר הסיכונים הנמצאים במעקב
- לרוב נדרש לדווח תקופתית (לפחות אחת לחודש) על סטאטוס בסיס הנתונים להנהלת הפרויקט.

שיטות אימות ותיקוף

- סקר עמיתים: החלטות התכן המערכתיות
- סקרי עמיתים עם הקבוצות המקצועיות הרלבנטיות לכל מכלול:
 - המודל הפונקציונאלי עבור כל מכלול (משימה, מאפיינים טכניים, פונקציות עיקריות)
 - "המכלול בסביבתו" – ממשקים חיצוניים ופריטי המידע
 - תפיסת הפעלה כללית של המכלול
- דוחות הנדסיים, אנליזות וסימולציות המצדיקים את ההקצאה של:
 - המאפיינים הטכניים לכל מכלול
 - הפונקציות לכל מכלול
 - הדרישות הלא-פונקציונאליות (Constraints) העיקריות לכל מכלול

- אימות ותיקוף המודלים המעודכנים
 - בדיקת קיום מודלים כללים לרמת המכלולים:
 - קיום מודל פונקציונאלי בבסיס הנתונים עבור פונקצית העל של המכלול (כולל פריטי המידע)
 - קיום מודל פיסיקלי-קונספטואלי בבסיס הנתונים עבור המרכיב/אלמנט (Component) של המכלול
 - בדיקה סטטית של המודל הפונקציונאלי (בדיקת אימות ועקביות)
 - בדיקה דינמית של המודל הפונקציונאלי (סימולציה של התרחישים)

בדיקה סטטית של המודלים

- מיפוי מלא בין הפונקציות ברמת העל לבין הפונקציות המוקצות למרכיבים
- תהליך ניתוח פעילויות המכלולים:
 - בדיקת כל הפעילויות (הפונקציות) הנדרשות לקליטת כל הכניסות וליצירת כל היציאות עבור כל מכלול
 - ניתוח פעילויות המערכת:
- בדיקת תאימות בין זרימות האינפורמציה לבין הפעילויות עבור כל מכלול
- בדיקת צימודים בין המכלולים בעזרת תרשימי N^2 - ואם יש צורך לשנות מיקום של פעילויות כדי להפחית בצימודים.

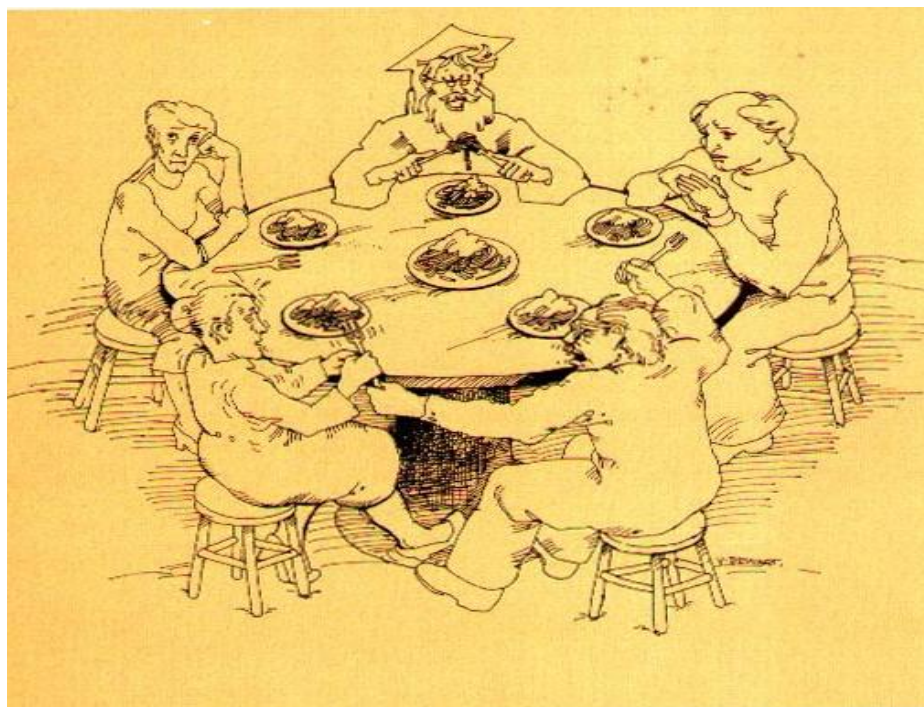
בדיקת דינמית של המודלים

- **סימולציות** מורחבות של התרחישים המערכתיים תוך שימוש בתרחישים של המכלולים
- סימולציה של **מצבים ומאורעות** ברמת העל, בעזרת מצבים ומאורעות של המרכיבים.
- בדיקת **עקביות** תהליכי על
- **אנליזות** להערכת ביצועים מערכתיים תוך שימוש בביצועים הנדרשים מהמכלולים
- יש להשיג **הסכמה עם הלקוח** לגבי כמה שיותר תרחישי ייחוס (תרחישים אלו ישמשו גם במהלך בדיקות הקבלה של המערכת).

בעיות קריטיות הקשורות לביצוע התרחישים

- **Deadlock (קיפאון)** - A **deadlock** is a situation wherein two or more competing actions are [waiting for the other to finish](#), and thus neither ever does.
- **Livelock (התנדדות)** - A **livelock** is similar to a deadlock, except that the states of the processes involved in the livelock [constantly change](#) with regard to one another, none progressing
- **Starvation (הרעבה)** - **Starvation** is a [multitasking](#)-related problem, where a [process](#) is perpetually denied necessary [resources](#). Without those resources, the program can never finish its task
- **Race (מירוץ)** - A **race condition** or **race hazard** is a flaw in a [system](#) or process whereby the output and/or result of the process is unexpectedly and critically dependent on the sequence or [timing](#) of other events. The term originates with the idea of two [signals](#) racing each other to influence the [output](#) first.

בעיית הפילוסופים הסועדים



בעיית הפילוסופים הסועדים היא המחשה

לבעיות תזמון ו**סנכרוניזציות** שמופיעות

בהקשרים של **עיבוד מקבילי** - מספר

תוכניות שרצות יחד וחולקות משאבים

משותפים.

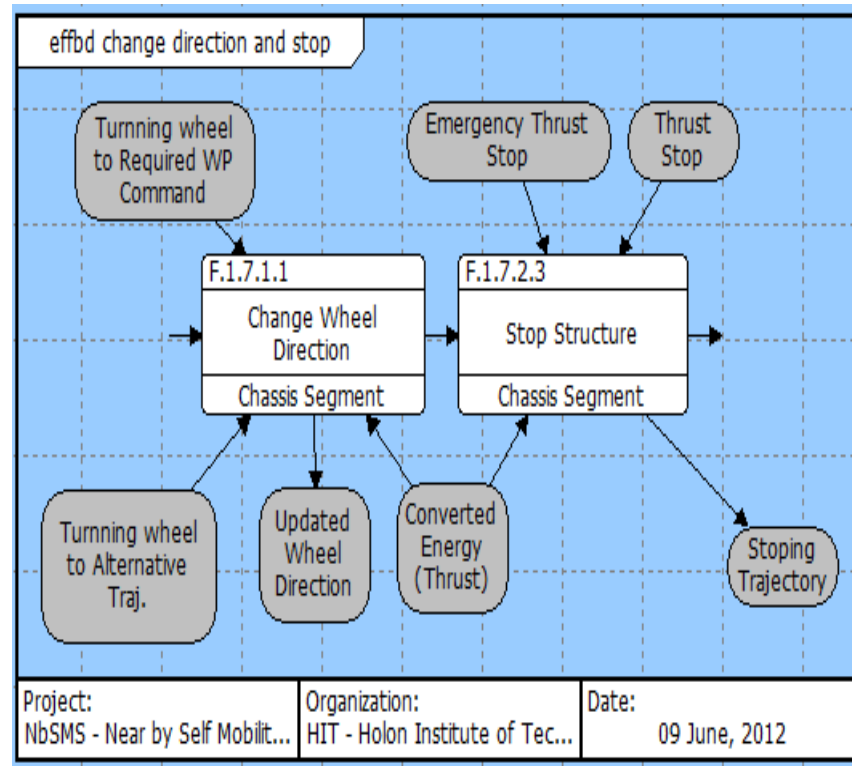
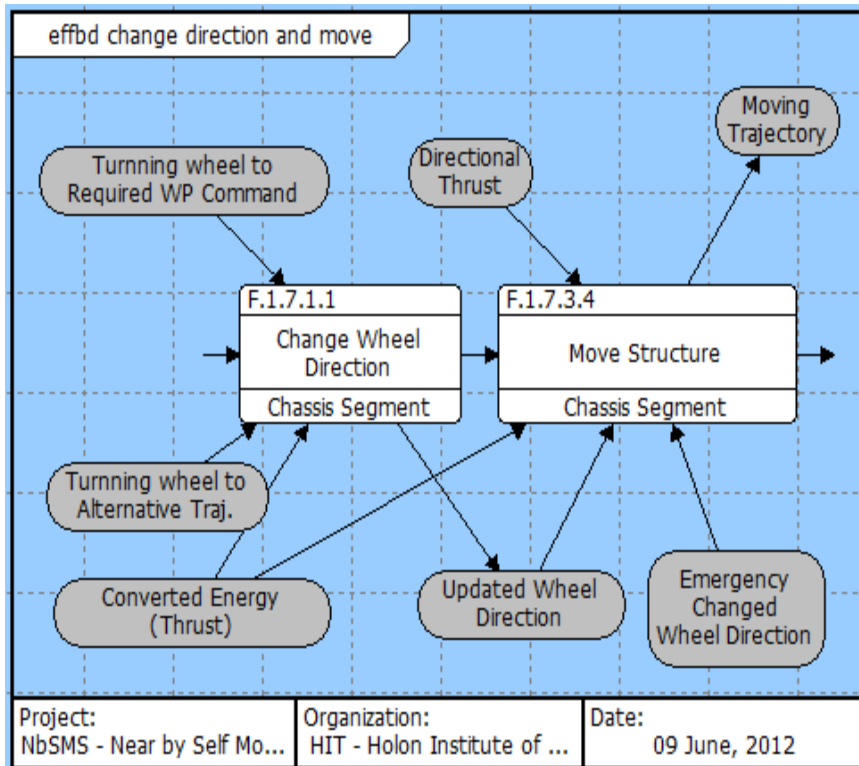
הבעיה נוסחה בשנת **1971**, בידי **אדסחר דייקסטר**.

בעיות תזמון (סינכרון)

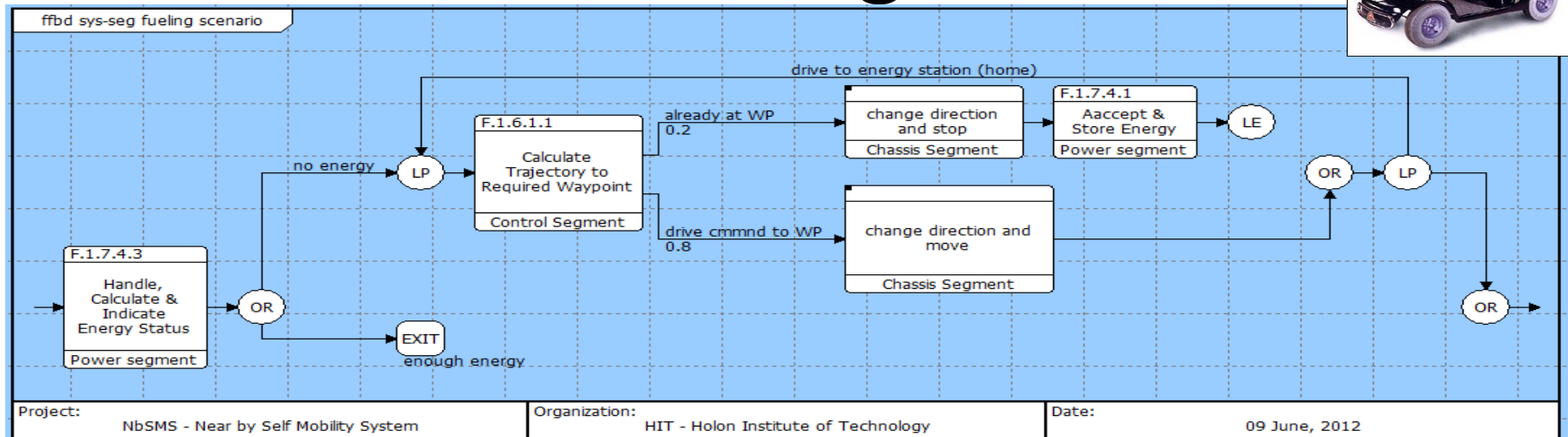
קיפאון - Deadlock	הרעבה - Starvation	מרוץ - Race	התנדנדות - Livelock
מצב בו מספר תהליכים מתחרים מחכים כל אחד לסיומו של האחר, ומכיוון שכך, <u>אף אחד</u> מהם אינו מסתיים.	מצב שבו נמנעת גישה של תהליך מסוים לאחד ממשאבי המערכת כך <u>שתהליך זה</u> לעולם לא יצליח לסיים את משימתו.	מצב שבו הרצה של מספר תהליכים במקביל תתבטא בתוצאות שונות כתוצאה מתזמון שונה	מצב שבו מספר תהליכים מתנדנדים בין מצבים שמפריעים אחד לשני להתקדם
<u>דוגמה</u> : שני אנשים עומדים בפתח של דלת, וכל אחד מהם מציע לרעהו את הזכות להיכנס ראשון. אם יתמידו בגישתם זו, לא יעברו לעולם בדלת	<u>דוגמא</u> : אדם רעב בא לקנות אוכל בחצות. היות והחנות סגורה, הוא חוזר לאחר 24 שעות וכך הלאה.	<u>דוגמא</u> : שני משתמשים משנים קובץ משותף באותו זמן. (הראשון שיעשה Save ישנה את הקובץ)	<u>דוגמא</u> : שני אנשים עומדים בפתח הדלת, כל אחד מהם עושה צעד קדימה, פותח הדלת ומחכה לשני שיעבור. כאשר השני אינו עובר – הראשון סוגר הדלת וחוזר אחורנית.



Chassis segment scenarios



SYS-SEG fueling scenario



Simulator for "sys-seg fueling scenario" (University Edition)

File Edit View Control Tools Help



Scale Control

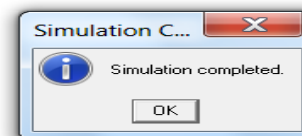
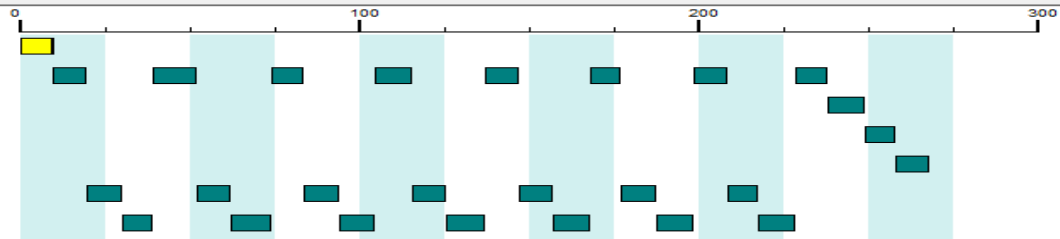
Normal

Time

268.11

Edit Order Recompute Order

- F.1.7.4.3 Handle, Calculate & Indicate Energy Status
- F.1.6.1.1 Calculate Trajectory to Required Waypoint
- F.1.7.1.1 Change Wheel Direction
- F.1.7.2.3 Stop Structure
- F.1.7.4.1 Aaccept & Store Energy
- F.1.7.1.1 Change Wheel Direction
- F.1.7.3.4 Move Structure



תפוקות ותוצרים ל- CDR מערכת

תפוקות תהליך התכן ההנדסי	תוצרי תהליך התכן ההנדסי
• מסמך תכן - SSDD • מסמך ממשקים פנימיים (בין המכלולים) – IRS	• דו"חות הצדקה טכניים להחלטות התכן המערכתיות • תוצאות בדיקות האימות השונות • בסיס נתונים מעודכן

תכולת מסמך התכנן המערכתי - SSDD

- documents system-wide design decisions
 - describes the proposed architectural design of the system
 - describes the system's concept of operation
 - lists the allocation of the system requirements to its components
-
- Is prepared by the system engineering, with the help of the software and other specialty engineering groups

SSDD - System / Subsystem Design Description

תהליך התכן ההנדסי ותוצריו

טבלאות בעלי עניין
ומסמכים מחייבים

דרישות "לקוח" (CRs)
(לקוח / בעלי עניין)

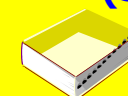
דרישות "בסיס" (BRs)



BRs Baseline

- מסמך דרישות בסיס (SRD)
- מסמך דרישות תפעול (ORD)

דרישות "מערכת" (SRs)

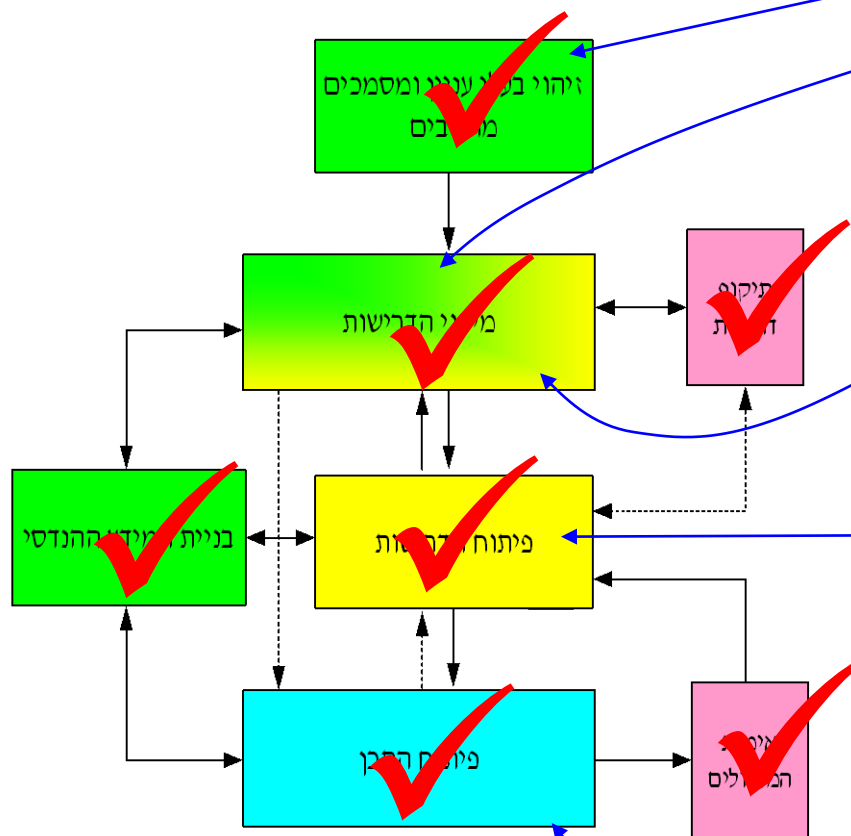


Functional Baseline

- מפרט תפעול (OCD)
- מפרט דרישות (SSS)
- מפרט ממשקים חיצוניים (IRS)

"דרישות" תכן

(דרישות "מקור" לרמה הבאה):
מסמך דרישות תכן (SSDD)
מסמך ממשקים פנימיים (IRS)



דרישות תכן חתומות – תחת נוהל שינויים רשמי

פרק 5

תהליך התכן ההנדסי או הנדסת דרישות

טבלאות בעלי עניין ומסמכים מחייבים

תהליך התכן ההנדסי ותוצריו

הגדרת הצורך

זיהוי בעלי עניין ומסמכים מחייבים

דרישות "לקוח" (CRs)
(לקוח / בעלי עניין)

דרישות "בסיס" (BRs)

- מסמך דרישות בסיס (SRD)
- מסמך דרישות תפעול (ORD)

מיצוי הדרישות

תיקוף דרישות

SRR

דרישות "מערכת" (SRs)

- מפרט תפעול (OCD)
- מפרט דרישות (SSS)
- מפרט ממשקים חיצוניים (IRS)

פיתוח הדרישות

PDR

דרישות "תכן" (DRs)

(דרישות "לקוח" לרמה הבאה):
- מסמך דרישות תכן (SSDD)
- מסמך ממשקים פנימיים (IRS)

פיתוח תכן

אימות המודולים

CDR

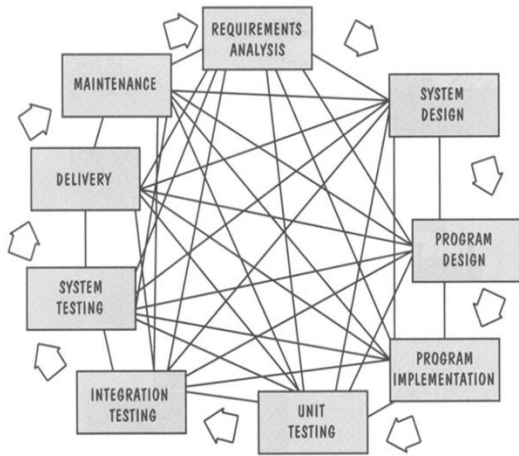
דרישות בסיס חתומות – תחת נוהל שינויים רשמי

דרישות מערכת חתומות – תחת נוהל שינויים רשמי

דרישות תכן חתומות – תחת נוהל שינויים רשמי

כל הזכויות שמורות

עקרונות תהליך התכן



כל שינוי בפרויקט
מחייב עדכון מסמכי
הדרישות.

- מתודולוגית תהליך התכן ההנדסי מבוססת על עקרון ההפשטה (Abstraction)

- האנליזה והאפיון של קומפוננטה (מערכת, מכלול, תת-מערכת...) הינם מדורגים ואיטרטיביים:

- עבור כל רמה מערכתית: מערך, מערכת, תת-מערכת, מכלול וכך הלאה מתבצע תהליך של תכן הנדסי:

□ דרישות לקוח,

□ דרישות בסיס,

□ דרישות מערכת,

□ דרישות תכן.

- עד לרמת המודולים שעבורם דרישות הלקוח מתורגמות למפרטי LRU ו-CSCI.

- בכל שלב, תוך כדי הקצאת/פירוק הדרישות לרמות (level) נמוכות יותר (allocation/decomposition) או תוך כדי שלבי התכן האחרים, אם מתגלה נושא שנשכח, או בעיה עם איזושהי דרישה, ניתן לחזור ולעדכן את ה-SRD ומפרט המערכת.



רמות של דרישות

קיימות 3 רמות של הגדרת דרישות:

- דרישה פונקציונלית/מבצעית: תאור צרכי הלקוח במונחים תפעוליים מבצעיים
 - **דרישת לקוח/דרישת בסיס** – המערכת תאפשר ניידות עצמית נוחה לאוכלוסיית היעד, במגוון מקומות ולמגוון מטרות, לפרקי זמן וטווח נדרשים.
- דרישה טכנית-מערכתית: פירוט הדרישה המבצעית במונחים טכניים ברי מדידה
 - **דרישת מערכת** – המערכת תהיה משוככת ובעלת הנעה עצמית לטווח של 80 ק"מ ולזמן פעולה רציף של 4 שעות.
- דרישה טכנית מפורטת: פירוט, **בשפת המפתח**, של דרישות האופיון, של תת-מערכת ההנעה (דרישת התכן מאפשרת הגדרה חד משמעית של תוצרי הפיתוח ובדיקתם):
 - **דרישת תכן** – על מכלול ההנעה לספק אנרגיה הנדרשת להסעת 500 ק"ג במהירות של 15 קמ"ש למרחק אופקי של 60 ק"מ.
 - .

ולסיכום - הנדסת/אנליזת דרישות

Requirements Analysis

המונח Requirement Engineering Analysis מתייחס לתהליכים הבאים:

- זיהוי וניסוח (Capture the Requirements) מהמערכת וכתובת מפרט דרישות מערכת SRD (מבצע מפרט מערכת SSS (טכני עם ערכים ברי-מדידה).
- ניסוח הדרישות צריך להיות (לא נדרוש משהו שאינו בר-השגה).
- תיקוף דרישות המערכת – בדיקה שאם עונה על צורך של הלקוח/המזמין.
- בדיקה שמימוש כל אחת מהדרישות לא יפגע בהם המערכתית (למשל לא נדרוש אמינות של מיליון שעות, זה יגרום לתאונות).
- בדיקה מעמיקה ומקיפה שאנו מבינים היטב את הדרישות ואת צורכיהם.
- הקצאת דרישות לתתי המערכות וכתובת מסמך התכנון המערכתי -

תהליך התכנון ההנדסי

סיימנו - בשעה טובה



הערות/הצעות לקורס הבא ???

נספח א

תוכניות עבודה

בניית תכניות עבודה

תנאי השוק

מחזור חיי
המערכת

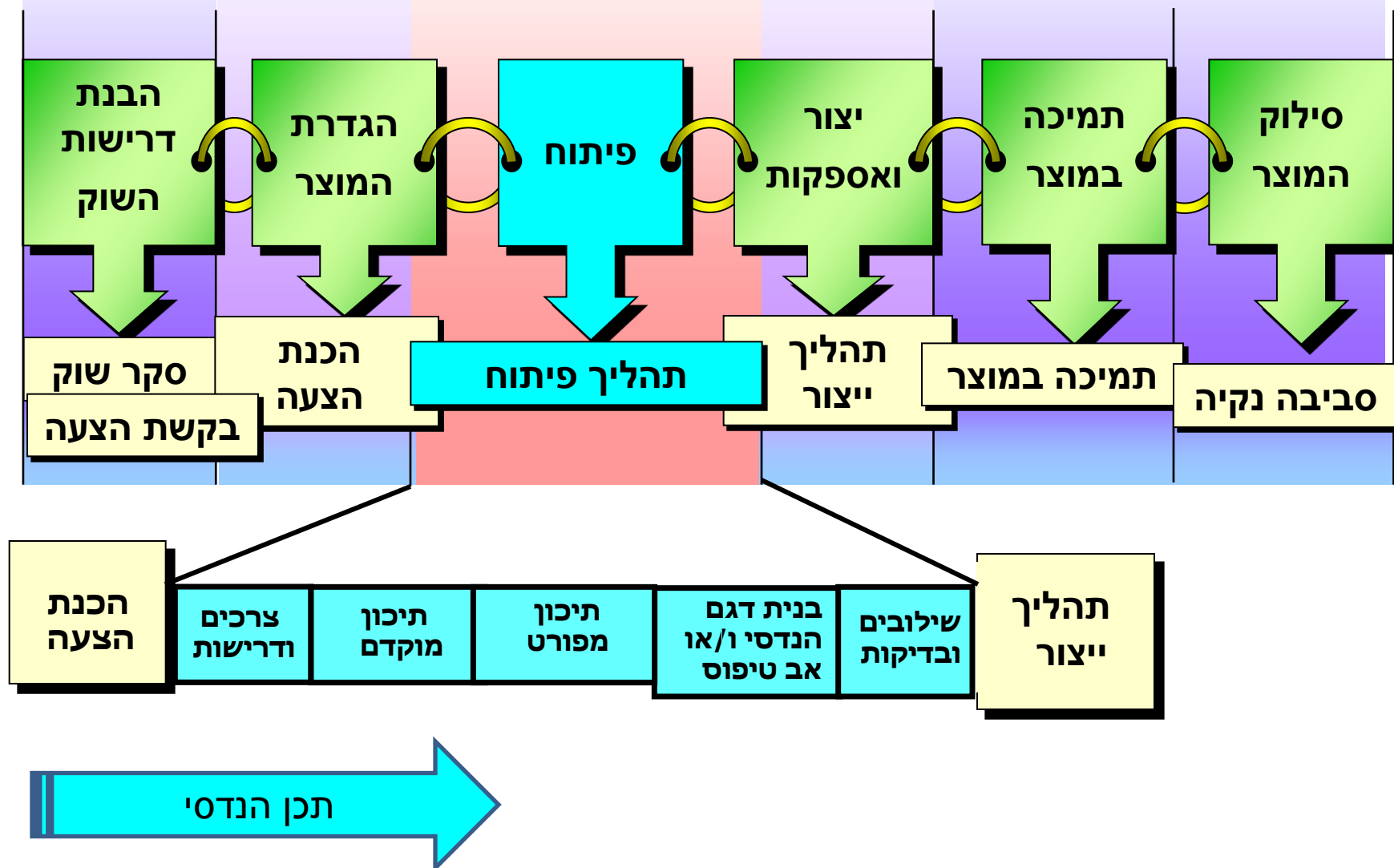
- סקלת בונן לפערי ידע
- תכניות להורדת סיכוני המערכת
- מסקנות מתהליך מיצוי הדרישות

תכנית עבודה ראשית

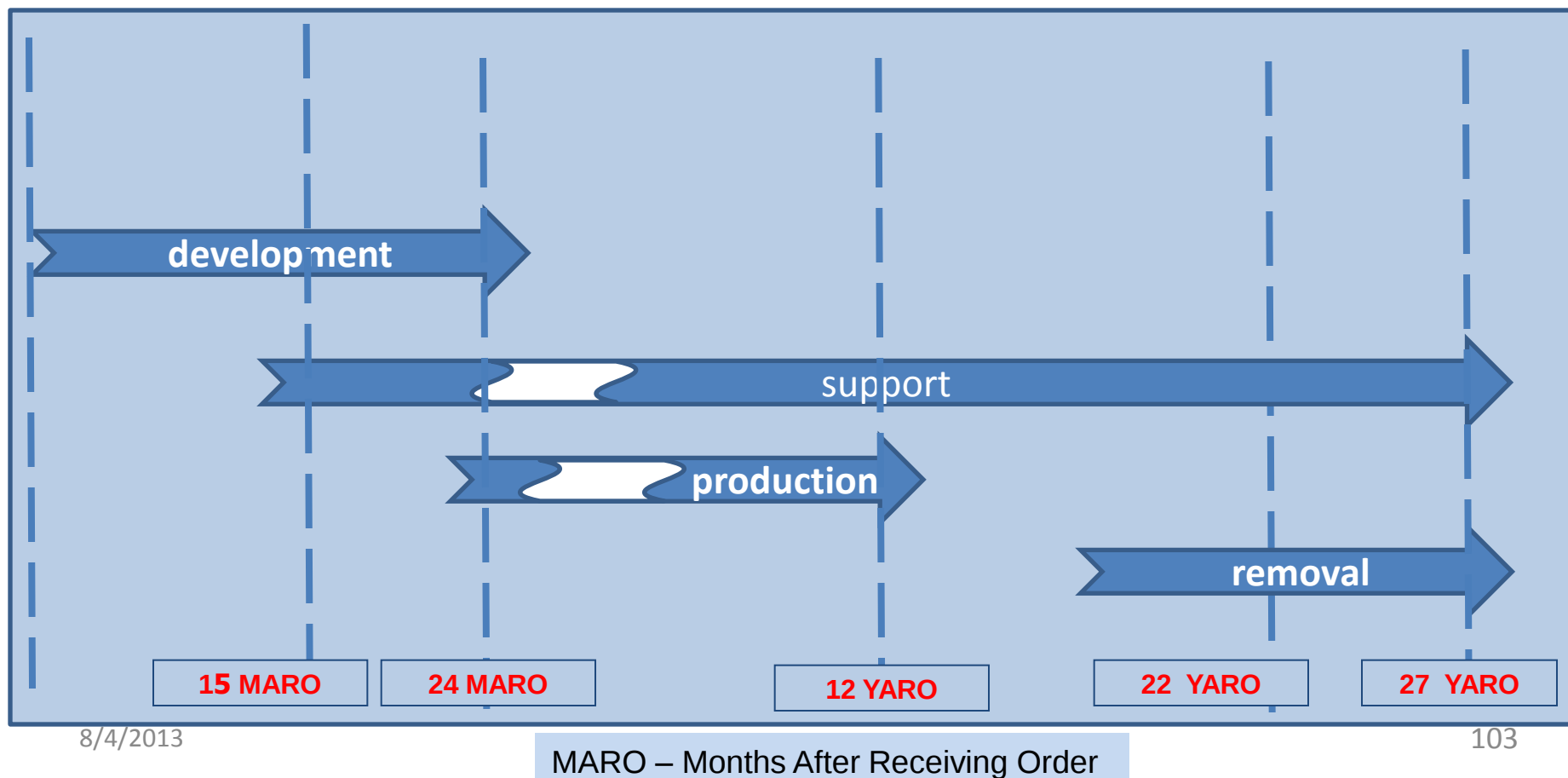
מודל פיתוח מערכת

תכניות עבודה

מחזור חיי מערכת



קביעת מחזור החיים – ת"ע ראשית

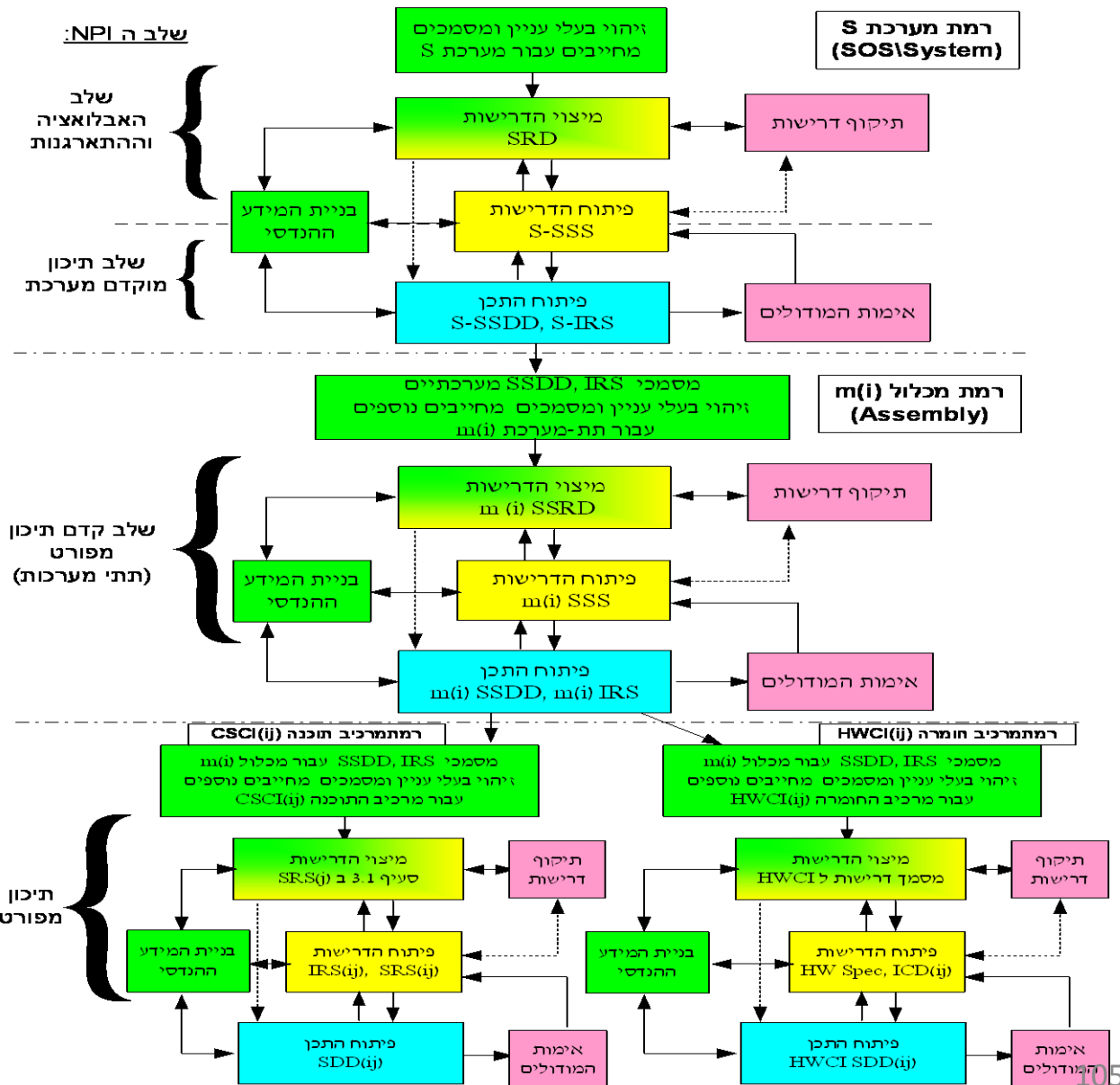
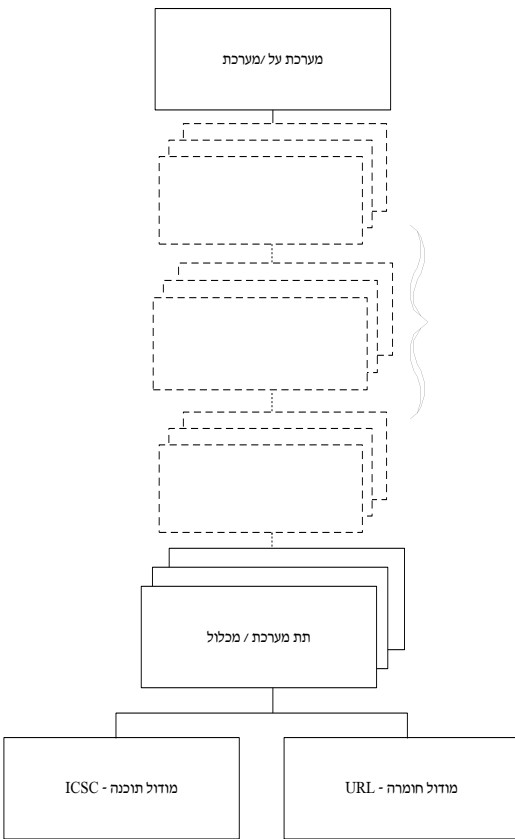


סקלת בון לפערי ידע

Level	Definition	Description	Development Cycles
1	Revision or Variant Design,	The project team is familiar with the solution (which has already been accomplished in-house), however small revisions are still required	1 to 1.5
2	Engineering Gap or Adaptive Design	The project team knows what to do and is familiar with the solution, however a full R&D effort is required	2 to 3
3	Original Design or Viability proof exists	The project team knows that a solution is feasible and that the technology exists, however the team does not know how to attain such a solution since it has never been attempted in-house before	M + 2 to 3
4	Research or No viability proof	The project team does not know whether or not a solution is possible or the technology available, research required	??? + M + 2 to 3

M = means Many cycles for level 3 components

תהליך תכן הנדסי במערכת



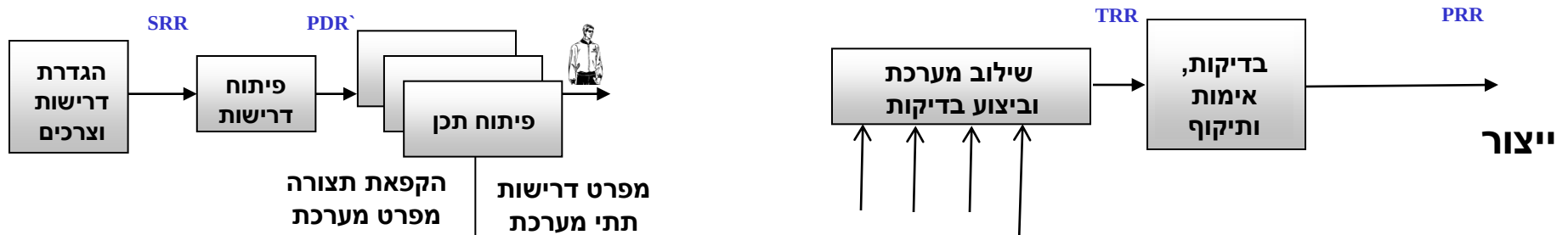
העברת התכן למהנדסי יישום (חומרה/תוכנה)



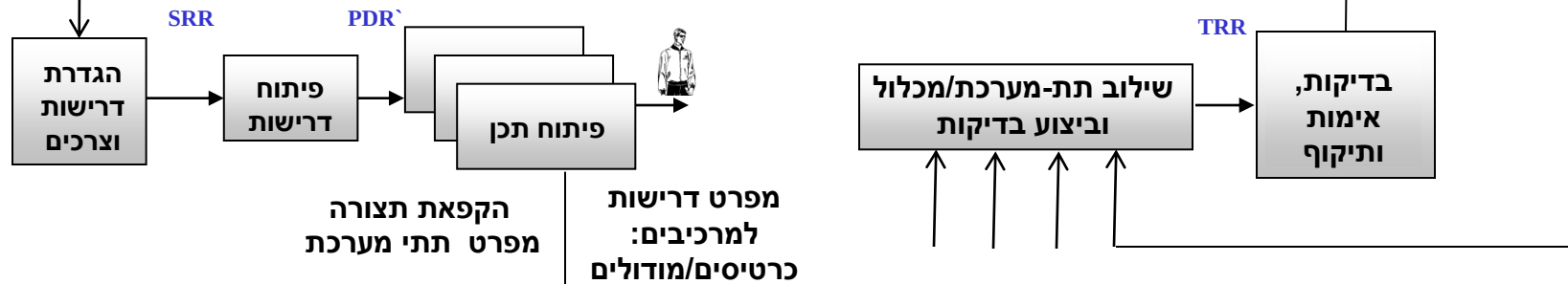
8/4/2013

תהליך פיתוח מפורט

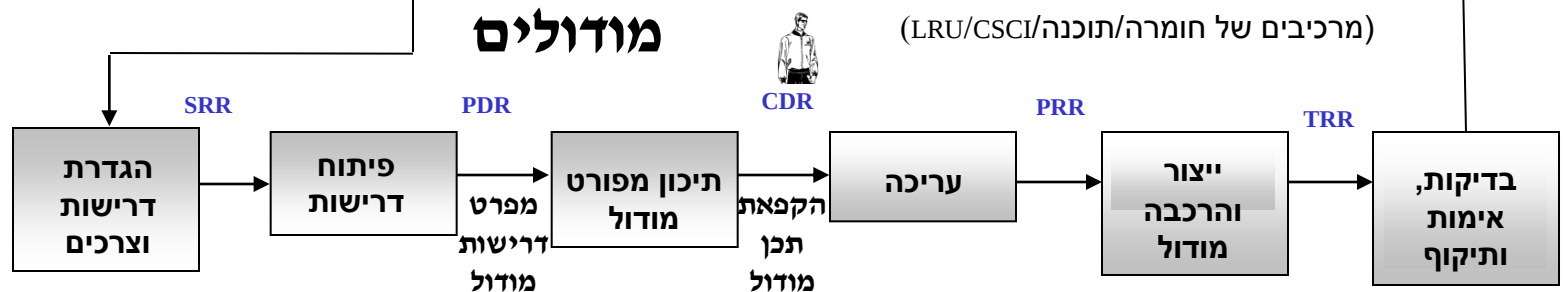
מערכת



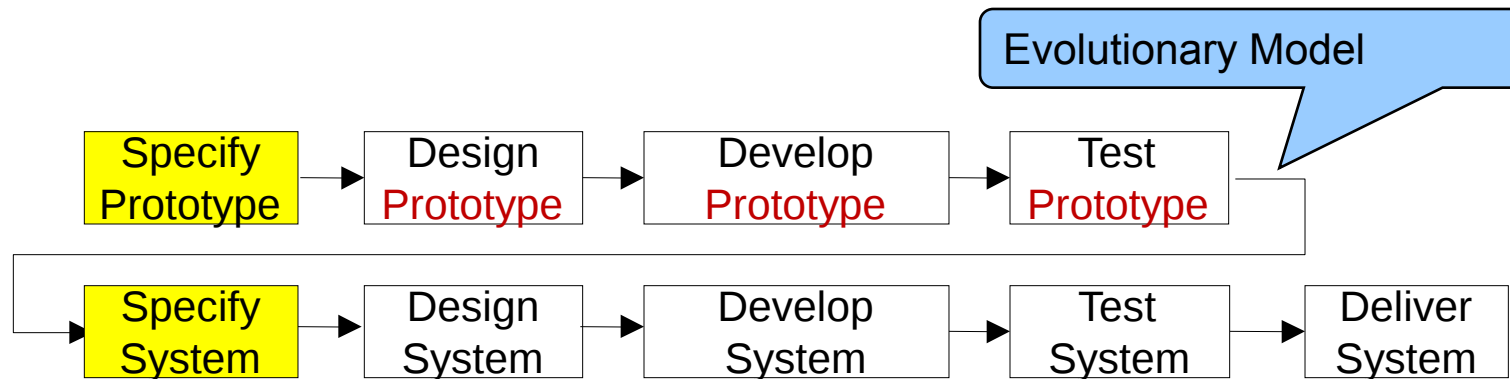
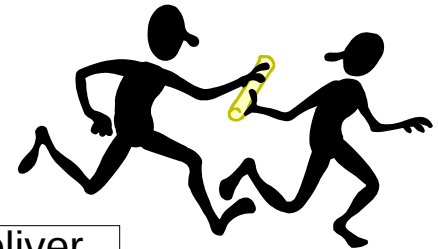
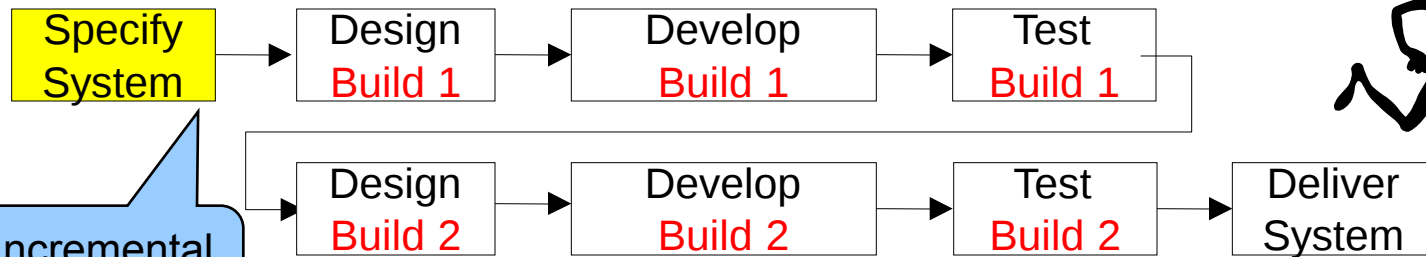
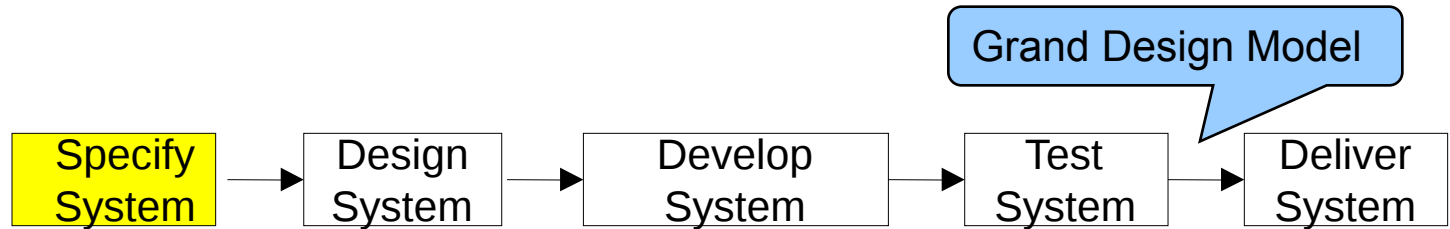
תתי-מערכת/מכלולים



מודולים



תיאור איסטרטגיות הפיתוח

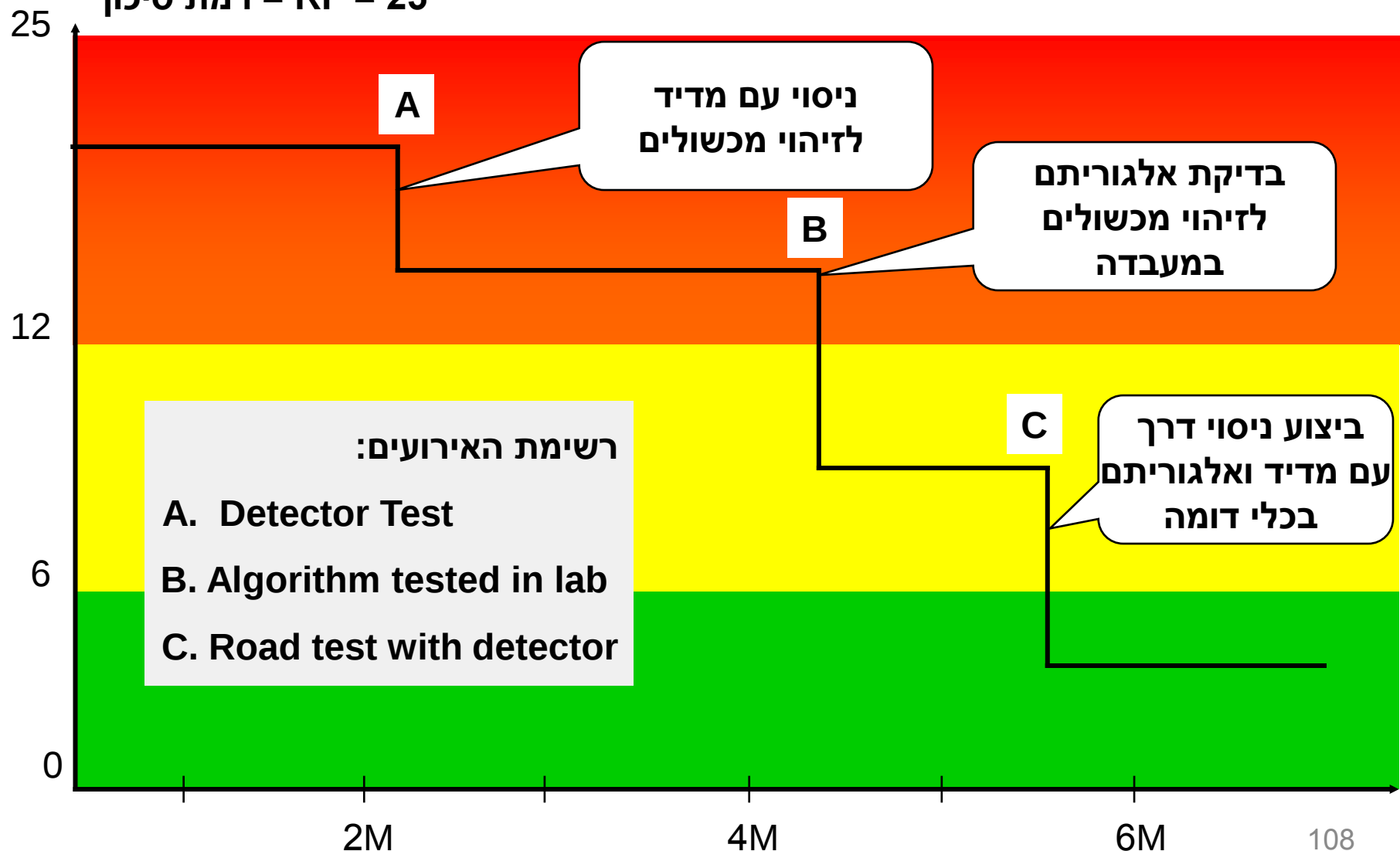


תכנית הורדת הסיכונים



שם הסיכון: זיהוי מכשולים ועצירה אוטומטית וחלקה

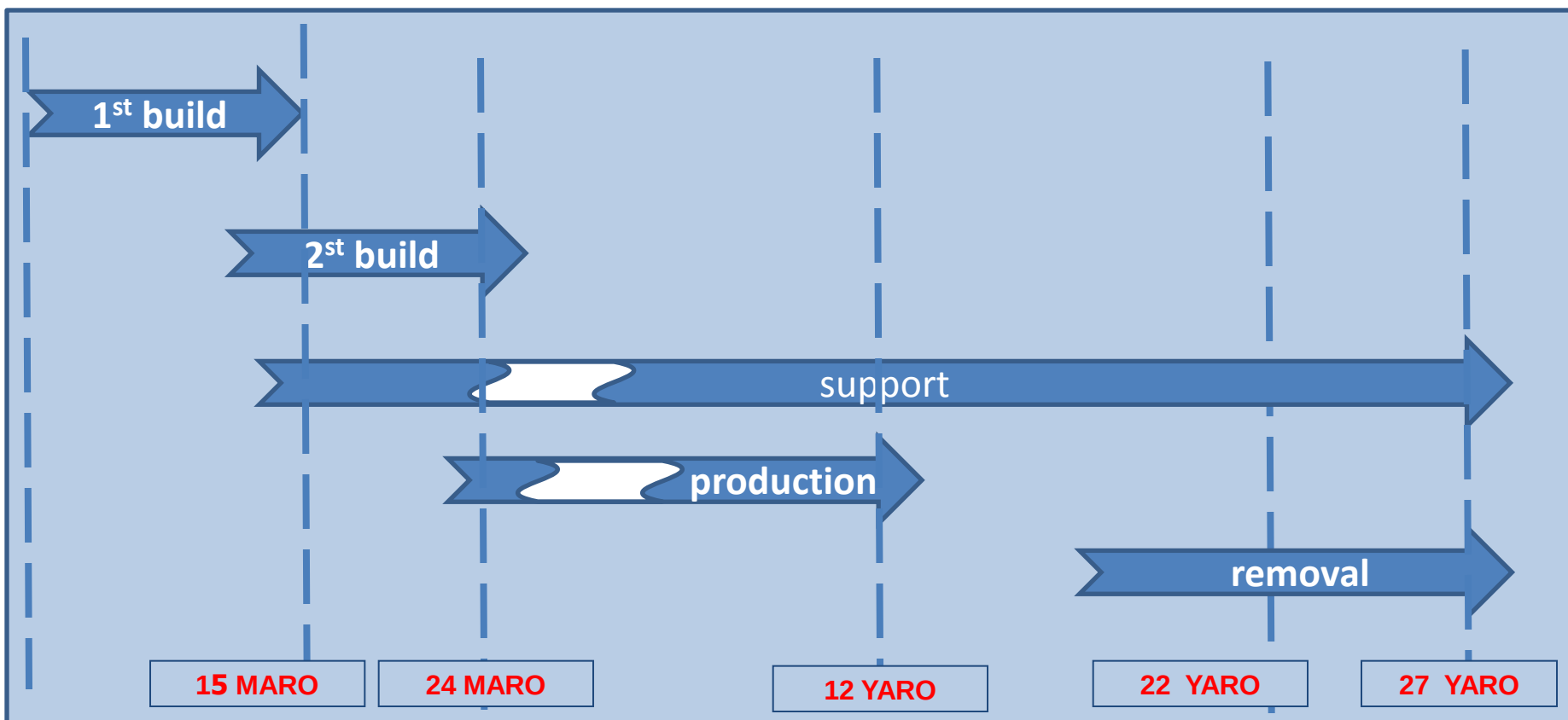
רמת סיכון = RF = 25



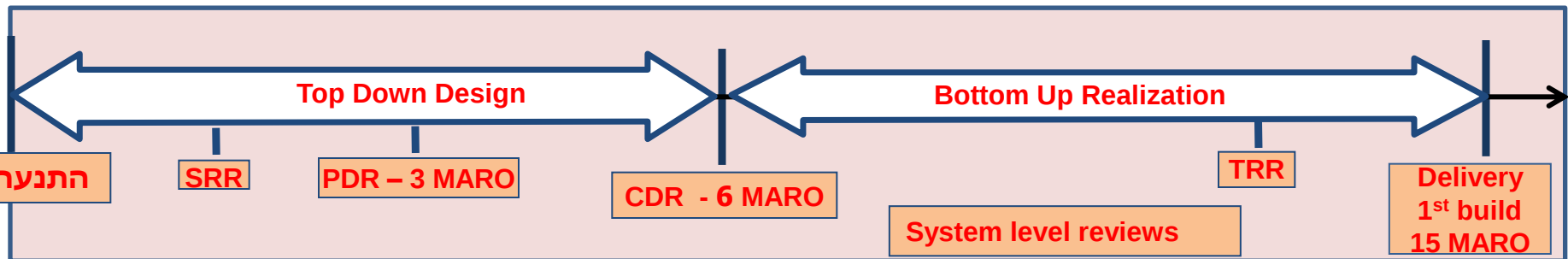
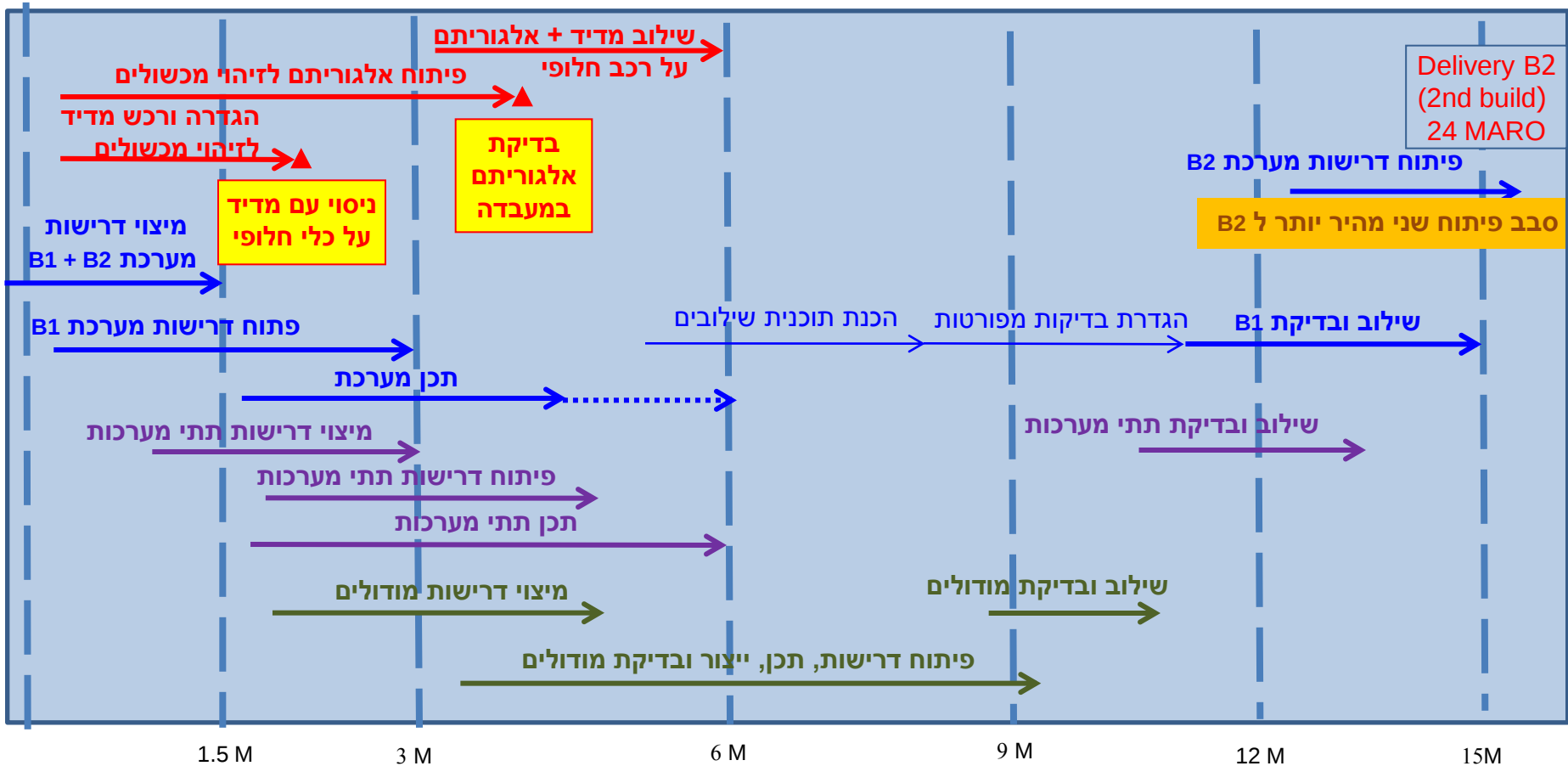
קביעת תהליך הפתוח – ת"ע ראשית



- ☐ פערי הידע לפי סקלת בון – רמה 2 <== נדרשים 2 סבבי פיתוח
- ☐ קיים סיכון אחד ברמה של 25 – זיהוי מכשולים ועצירה אוטומטית וחלקה
- ☐ יש להגיע עם מוצר לשוק תוך 24 חודש
 - ☐ Build 1 – ב-15 חודש
 - ☐ Build 2 – ב-24 חודש



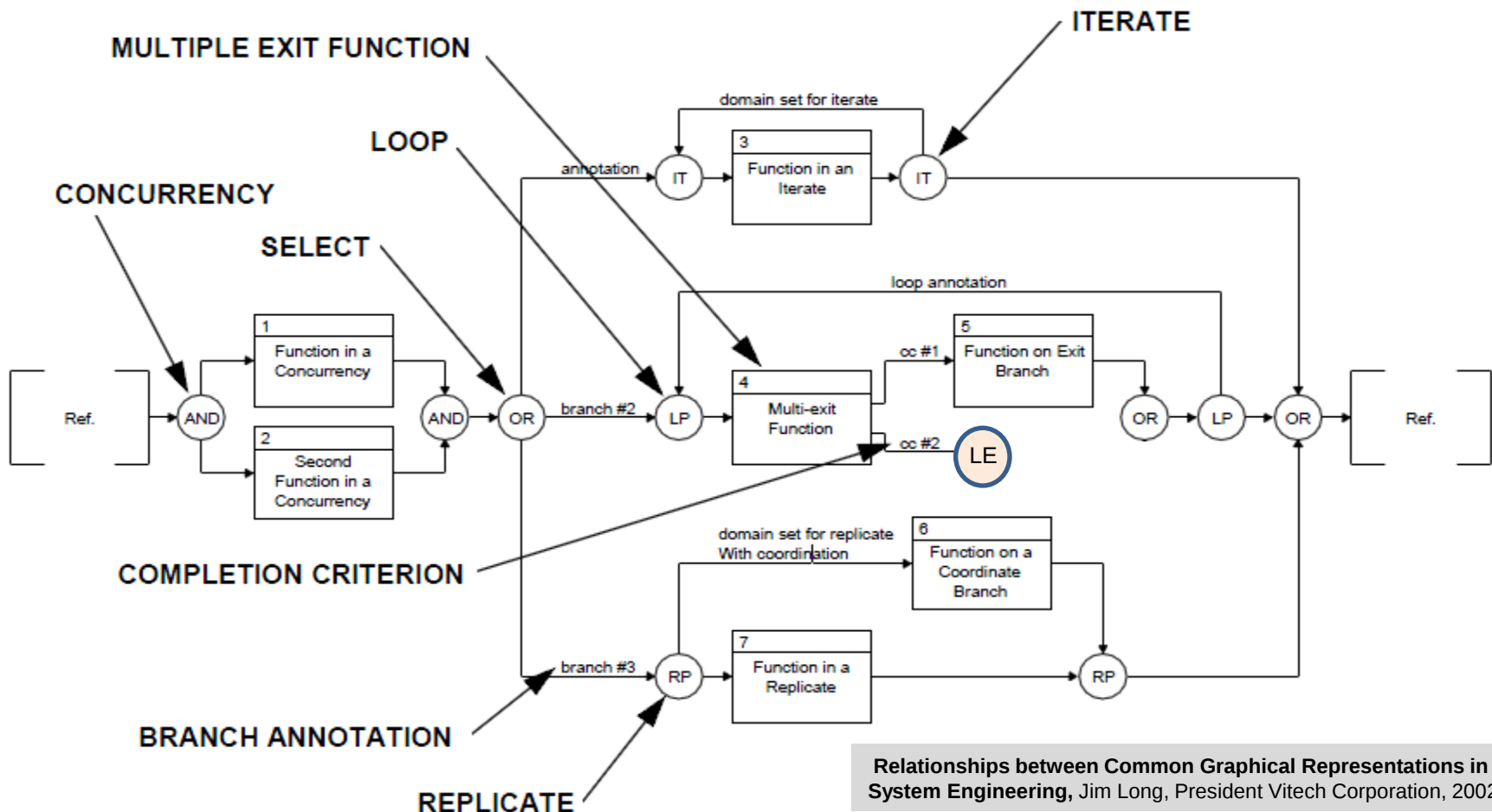
תכנית הפיתוח - טיוטא ראשונית



נספח ב

יכולת סימולציה של תרחישים

דיאגראמת FFBD

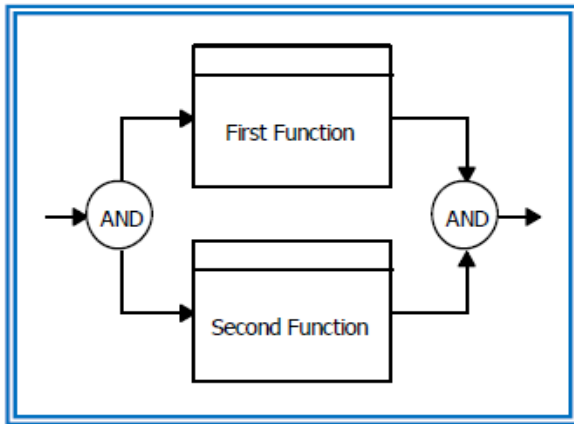


Relationships between Common Graphical Representations in System Engineering, Jim Long, President Vitech Corporation, 2002

Figure 2 – Control Constructs Available for Function Flow Block Diagrams

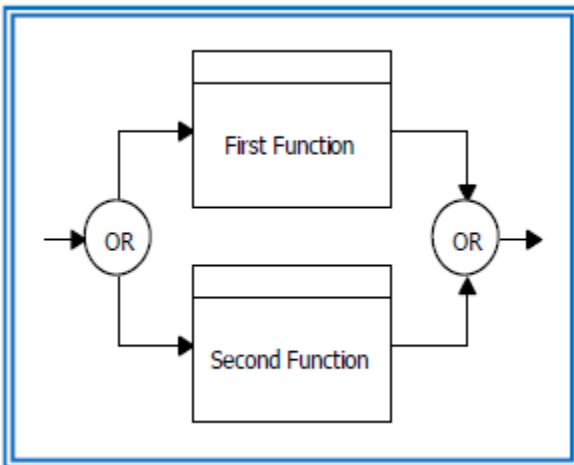
The FFBD presents only the control sequencing for the functions

תבניות בקרה (1)



- AND (תבנית מקבילה)

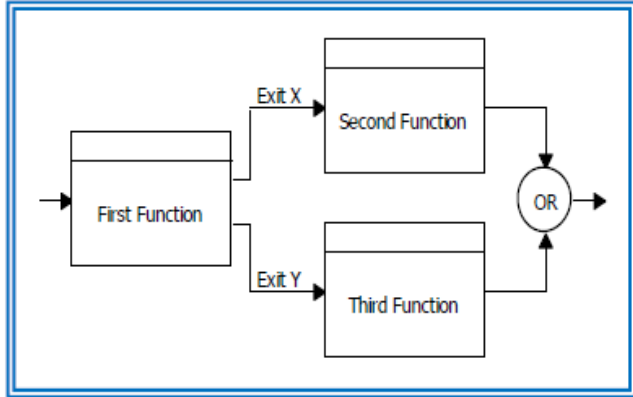
- כל ערוצי ה-AND מתבצעים במקביל.
- המעבר לתבנית הבאה יבוצע רק לאחר סיום כל ערוצי ה-AND



- OR (תבנית בחירה)

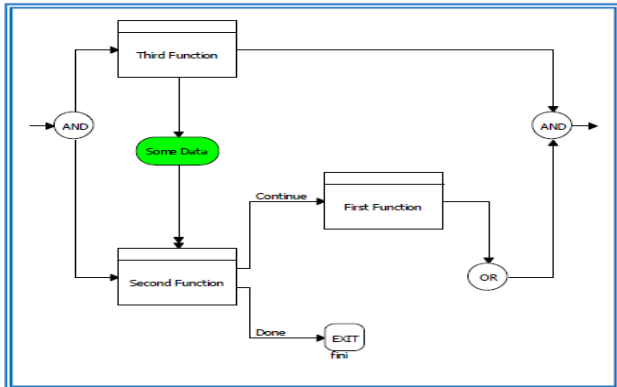
- רק ערוץ אחד מתבצע
- ניתן להגדיר הסתברויות שונות לערוצים השונים
- ללא הגדרה: לכל אחד מהערוצים – הסתברות שווה לביצוע.

תבניות בקרה (2)



• Multi-Exit Function (תפקוד רב-יציאות)

- התבנית מתקבלת ע"י הגדרת כמה יציאות מפונקציה אחת. היציאות מתאחדות בצומת OR
- כל יציאה מוגדרת בשם ויכולה להכיל פונקציות ותבניות בקרה אחרים.
- ניתן להגדיר הסתברויות לכל אחת מהיציאות.
- בד"כ משתמשים באפשרות הנ"ל עבור פונקציות החלטה (במקום בתבנית בחירה).



• Exit Node (צומת יציאה)

- צומת יציאה מייצגת השלמת ההתנהגות המפורטת של תפקוד רב-יציאות ואפשרות סיום שלה.
- רצוי שתהיה לפחות צומת יציאה אחת מכל ענף.
- כאשר הסימולטור נתקל בצומת יציאה – הוא עובר לרמה הגבוהה יותר של הסימולציה.

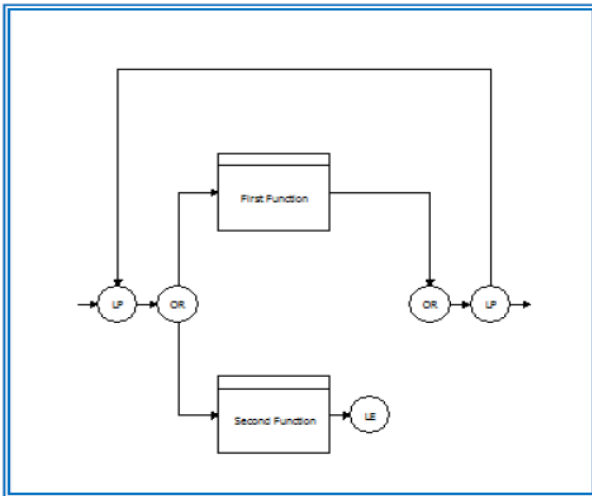
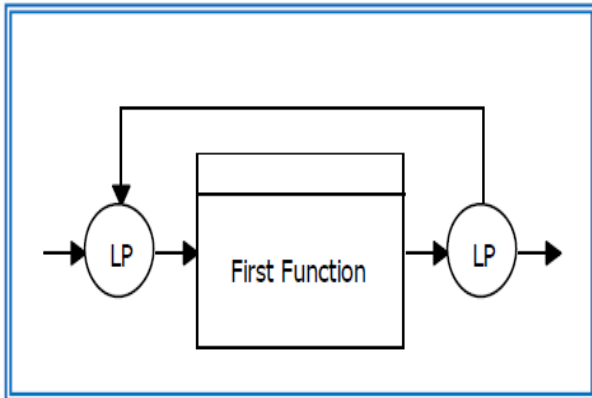
תבניות בקרה (3)

- לולאה (Loop)

- לולאה כוללת 2 צמתי NP שביניהם ענף היכול לכלול תבניות בקרה שונות.
- בדרך כלל הלולאה כוללת מוצא (Loop Exit) המגדיר את תנאי היציאה. (אחרת – מתקבלת לולאה אינסופית).
- רצוי להוסיף כיתוב תיאורי על הלולאה (Annotation).

- מוצא לולאה (loop Exit)

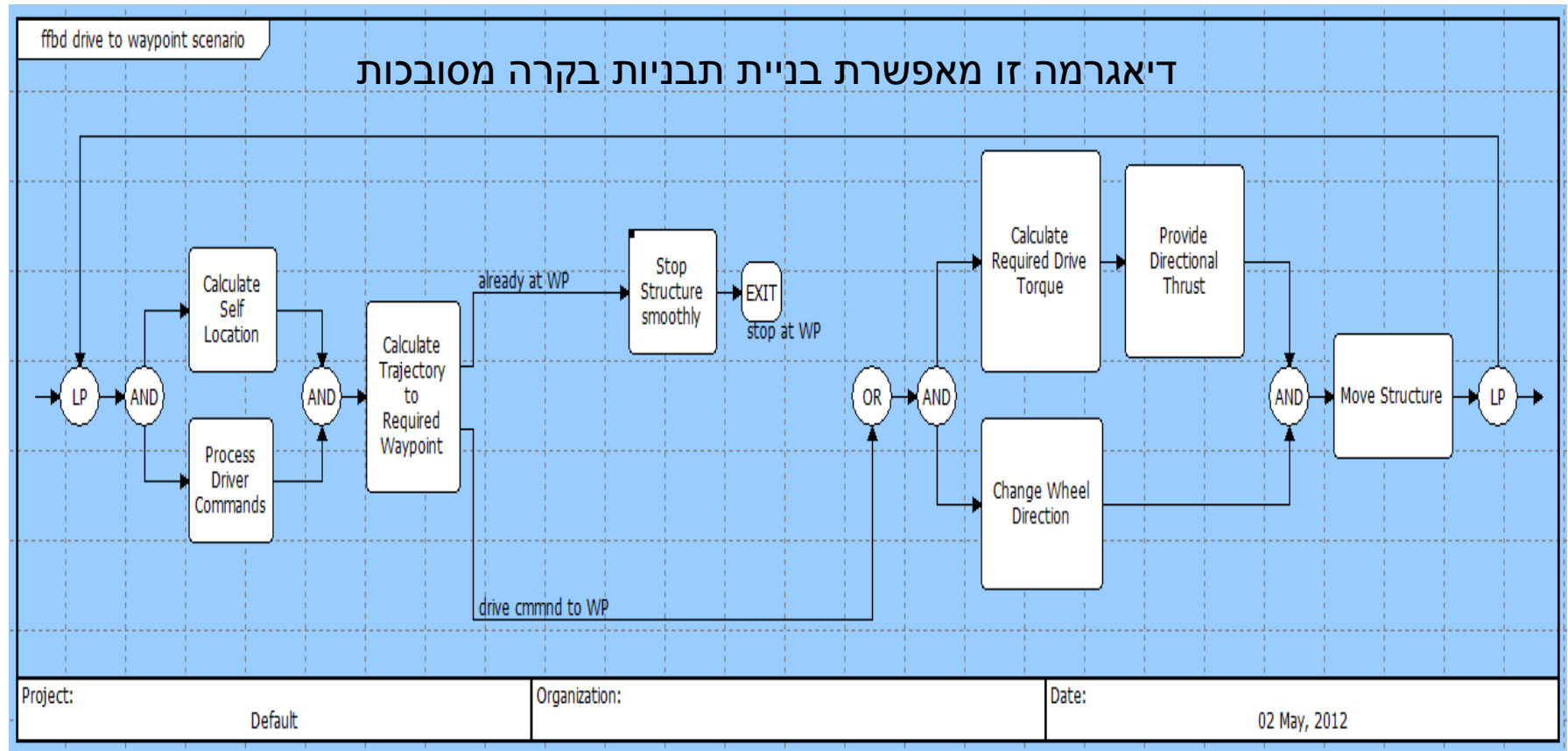
- כאשר הסימולטור מגיע לצומת המוצא – הלולאה הפנימית ביותר נפסקת, ומאפשרת המשך סימולציה.





דוגמא (*): דיאגרמת FFBD

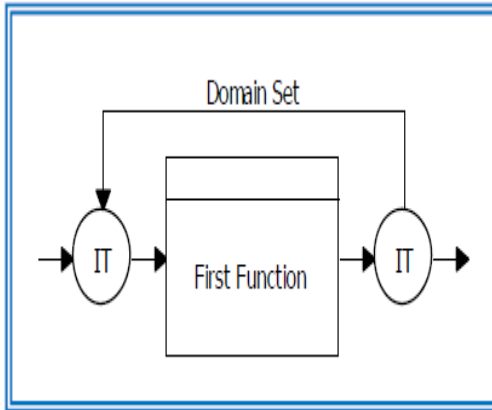
Functional Flows Block Diagram



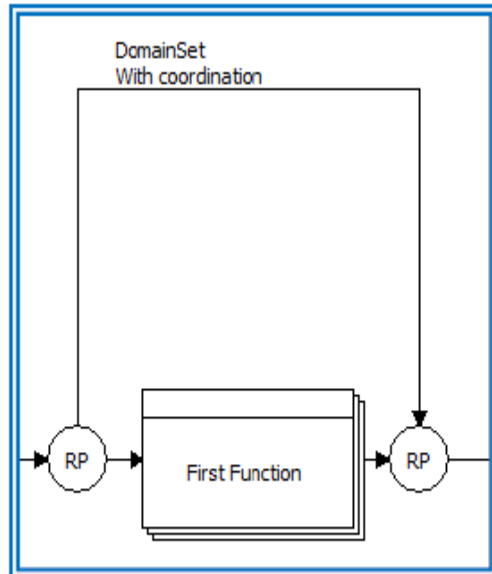
(*) – בסיס נתונים ישן

תבניות בקרה (4)

• תבנית חזרה (Iterate)



- תבנית חזרה כוללת 2 צמתי IT וביניהם ענף ולולאה חוזרת.
- הלולאה החוזרת מסומנת אוטומטית בשם שניתן לאלמנט ה- DomainSet.
- הענף יכול לכלול מבני בקרה שונים שיבוצעו כמספר הפעמים המוגדר במאפיין המספרי של ה- DomainSet. (מוגדרת ברירת מחדל למספר)



• תבנית העתקה (Replicate)

- תבנית העתקה כוללת 2 RP וביניהם ענף עיקרי וענף מתאם.
- הענף העיקרי כולל פעילויות זהות המתרחשות במקביל
- הענף המתאם מסומן אוטומטית בשם שניתן לאלמנט ה- domainSet. הוא כולל פונקציה המאמת את בין הפעילויות המתרחשות בענף העיקרי.

הוספת ערוץ kill לתרחיש

ערוץ ה-kill מאפשר קיום מאורע הגורם לסיום כל הפעילות המקבילה לו.

בדוגמא שלהלן: כאשר ה"פריט המעורר" some data יגיע לפונקציה Prepare Report, הרי שבגמר הבצוע שלה כל הפעילות המקבילה תיפסק והסימולציה תעבור ל-Stop work.

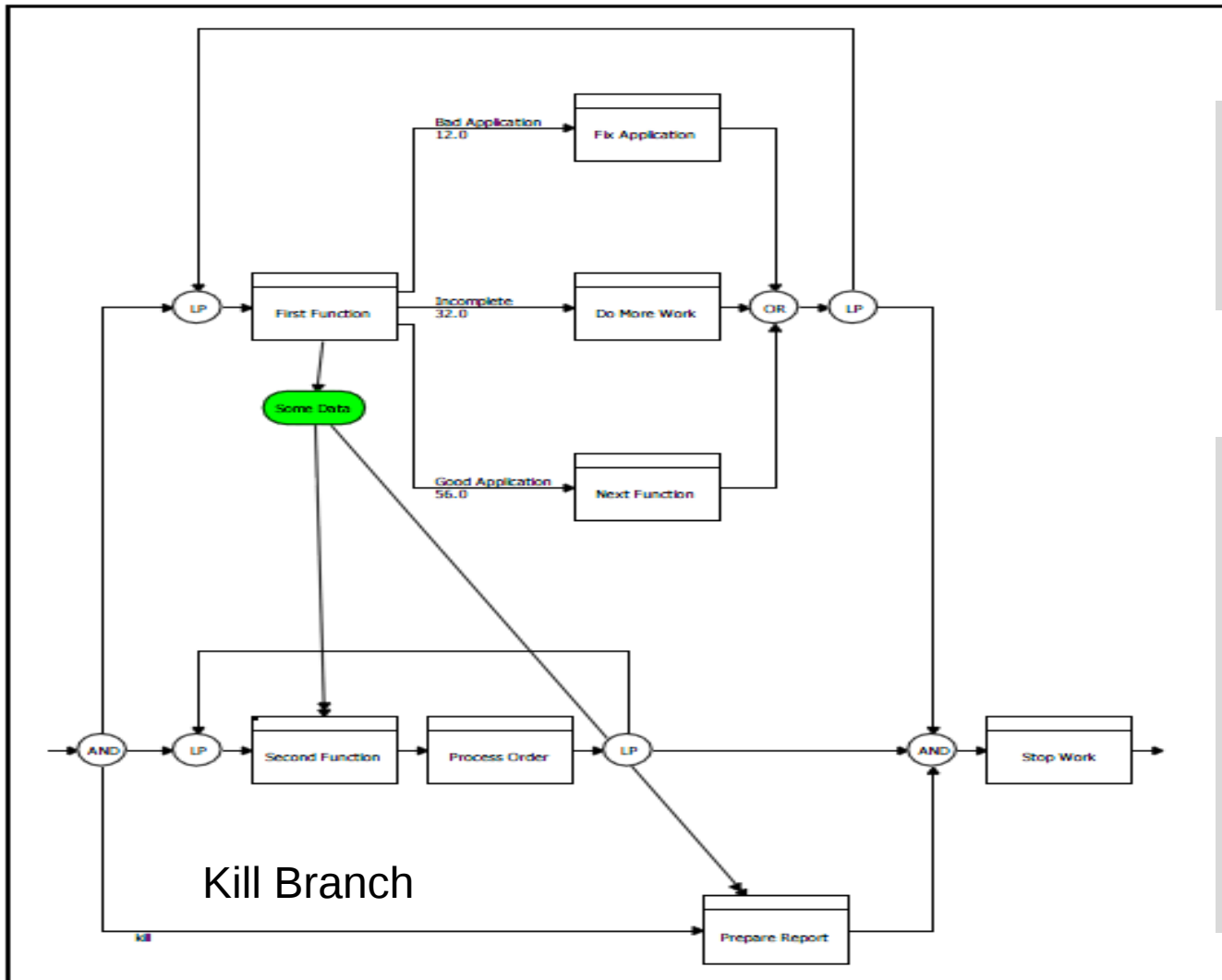


Figure 42 – Model with Kill Branch

דוגמא: דיאגרמת EFFBD

- הדיאגרמה כוללת:
- 1 – את הפונקציות
 - 2 – את תבניות הבקרה (זרימת הבקרה של התהליך)
 - 3 – את זרימת פריטי המידע (הדינמיקה המערכתית)

ניתן להגדיר לכל תפקוד זמן ביצוע/עלויות וכן להוסיף לתהליכים משאבים ומגבלות שימוש בהם לצורך סימולציות דינמיות.

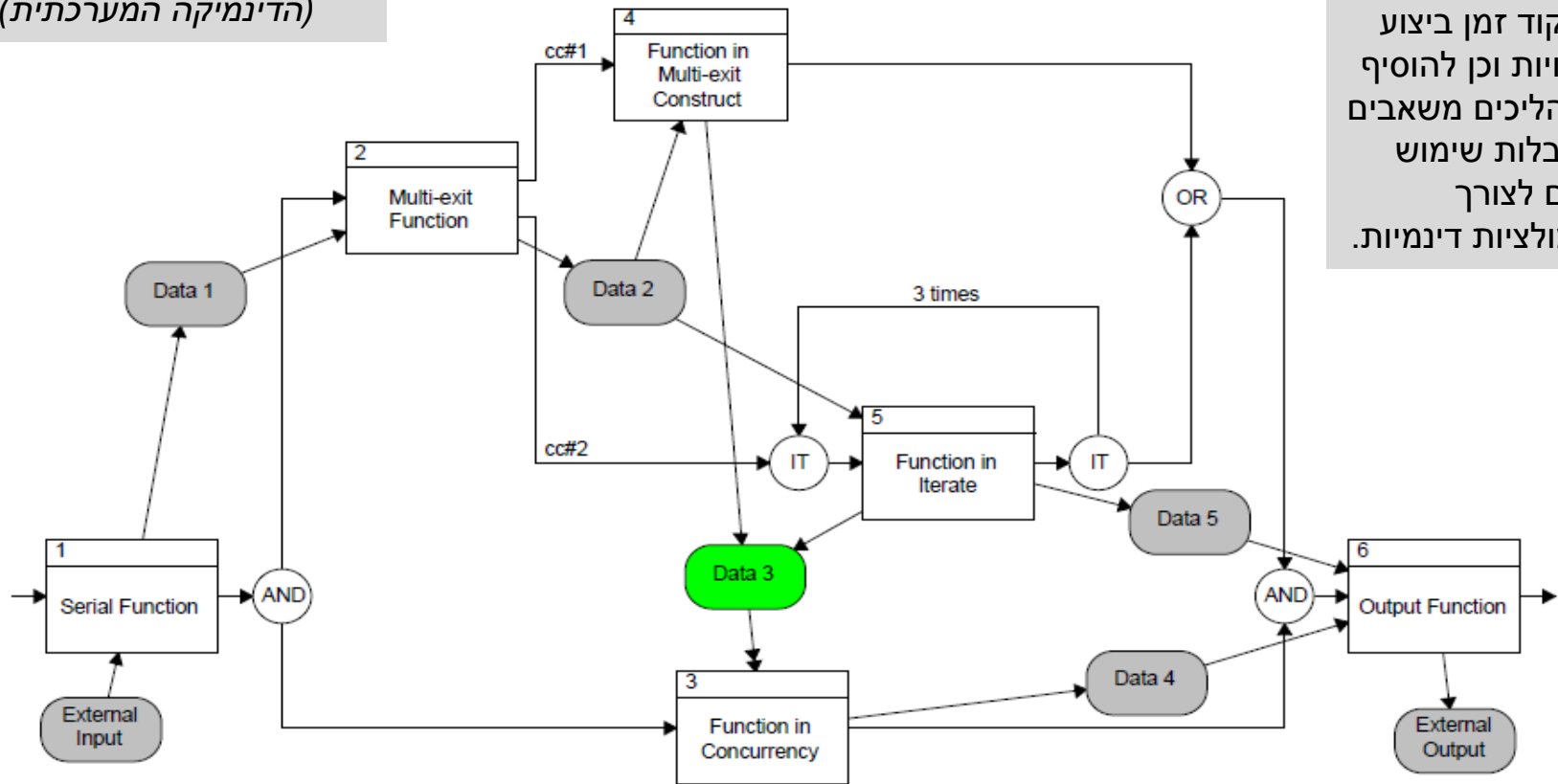


Figure 8 – Sample Enhanced FFBD

Relationships between Common Graphical Representations in System Engineering, Jim Long, President Vitech Corporation, 2002

פונקציה/תפקוד צריכה להיות מאופשרת ומעוררת לפני תחילת הביצוע

EFFBD Example

Enhanced Functional Flows Block Diagram



NbSMS – EFFBD for the functional context

EFFBD's also called behavior diagrams.

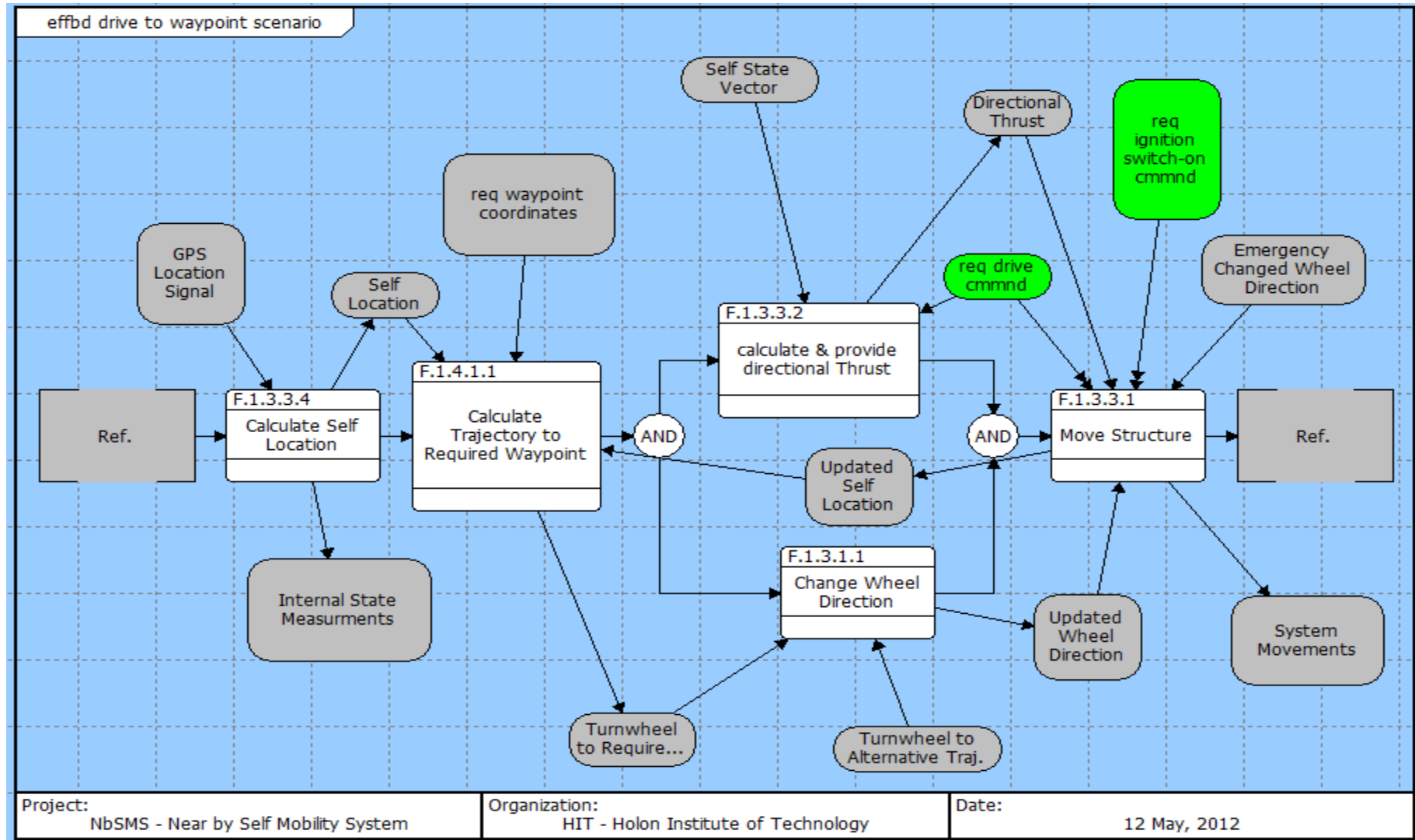
The EFFBD presents both:

- The control sequencing for the **functions**
- The data flow between the **functions**





דוגמא: דיאגרמת EFFBD



סימולציה הכוללת ערוצים פיסיים

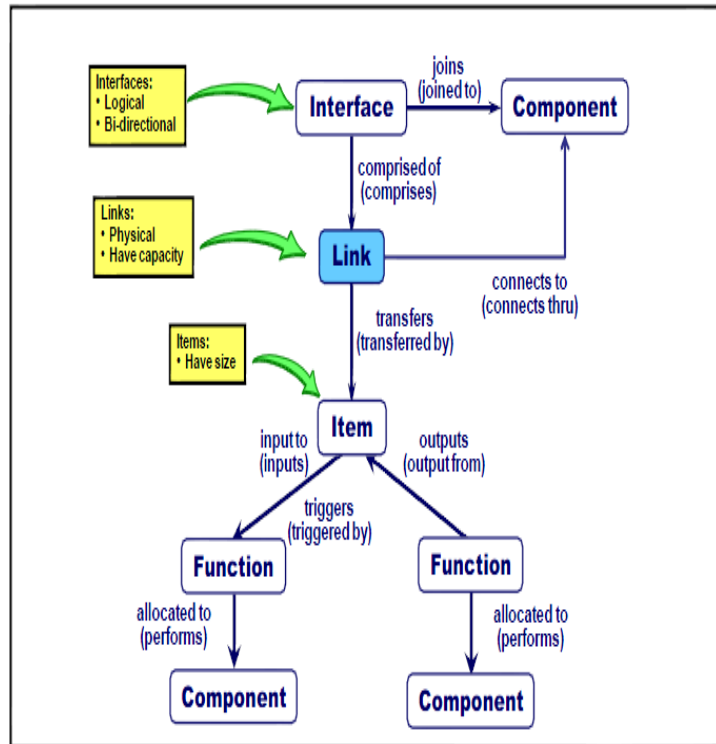


Figure 44 - Interface and Link Schema Diagram

- ניתן להגדיר מאפיינים:
- לממשקים הפיסיים (Links):
 - זמן השהייה
 - קיבולת
- ולפריט הזרימה (Item):
 - גודל
- הסימולטור מחשב את זמן המעבר של פריט הזרימה בממשק הפיסי:

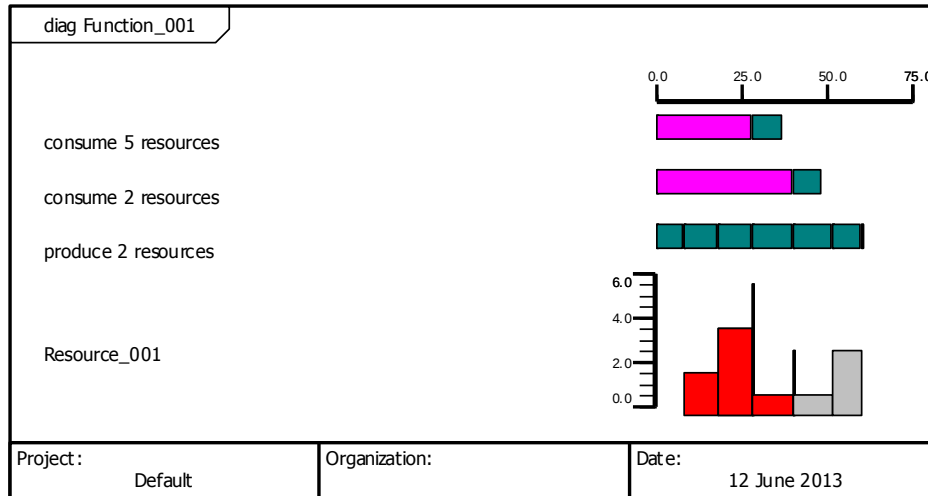
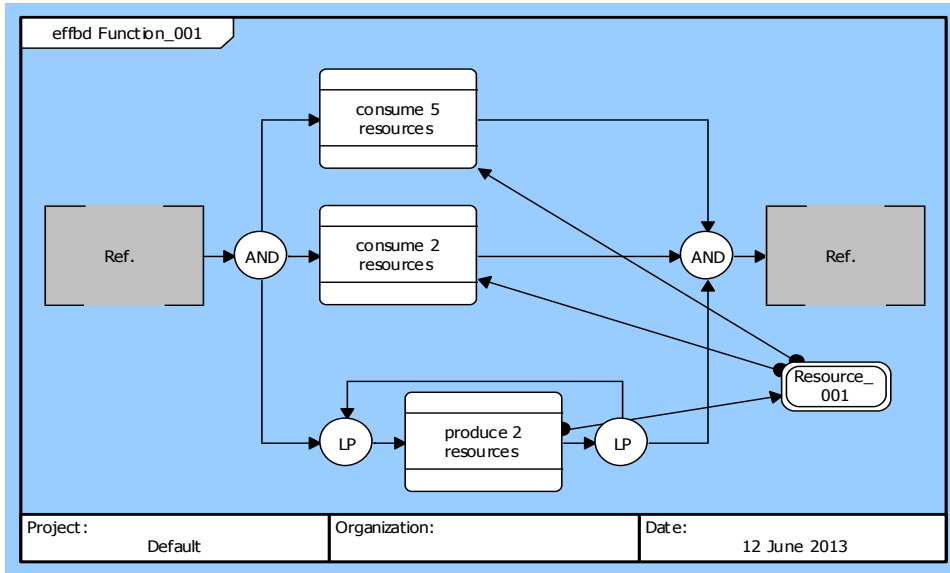
$$\text{Total Link time} = \text{link delay} + \text{Item size/link capacity}$$

ומוסיף את זה לזמן הנדרש לביצוע התרחישים

הוספת משאב לסימולציה (*)

- עבור "משאב" (Resource) ניתן להגדיר:
 - ערך התחלתי (initial amount)
 - ערך מירבי (maximum amount)
- פונקציה יכולה ליצור (produce) משאב: ניתן להגדיר את ערך המשאב הנוצר.
- ניתן להגדיר לכל פונקציה המשתמשת במשאב האם היא:
 - "תופסת" אותו - משאב "נצרך ומשתחרר" (captures ב- relationships) – לאחר שהפונקציה תסתיים המשאב "משתחרר" וחוזר לערכו ההתחלתי. בשרטוט ה- EFFBD - חץ דו-כיווני יחבר בין הפונקציה למשאב.
 - "צורכת" אותו - משאב "נצרך ומתכלה" (consumes ב- relationships) – לאחר שהפונקציה מסתיימת ערכו של המשאב ירד בערך הנצרך. בשרטוט ה- EFFBD - יופיע עיגול בצד של המשאב וחץ בצד של הפונקציה.
 - ניתן לקבוע מהו ערכו של המשאב "הנתפס" או ה"נצרך" ע"י הפונקציה.
- אם לפונקציה הוצב הערך:
 - TRUE ב- acquire available, היא תופסת את המשאב (שעדיין לא הגיע לכמות הנדרשת) עד שתוכל לרוץ. בזמן שהמשאב "תפוס", פונקציות אחרות לא יכולות להשתמש בו.
 - FALSE ב- acquire available, היא אינה תופסת את המשאב (שעדיין לא הגיע לכמות הנדרשת). במקרה זה אם תהיה פונקציה אחרת, לה הכמות הנתונה של המשאב מספיקה לריצה, היא תרוץ קודם.

שימוש במשאב (1)



❑ הערך TRUE הוגדר לפונקציה הראשונה (consume 5 resources) – כלומר היא תמתין ל-5 משאבים זמינים ואז תרוץ.

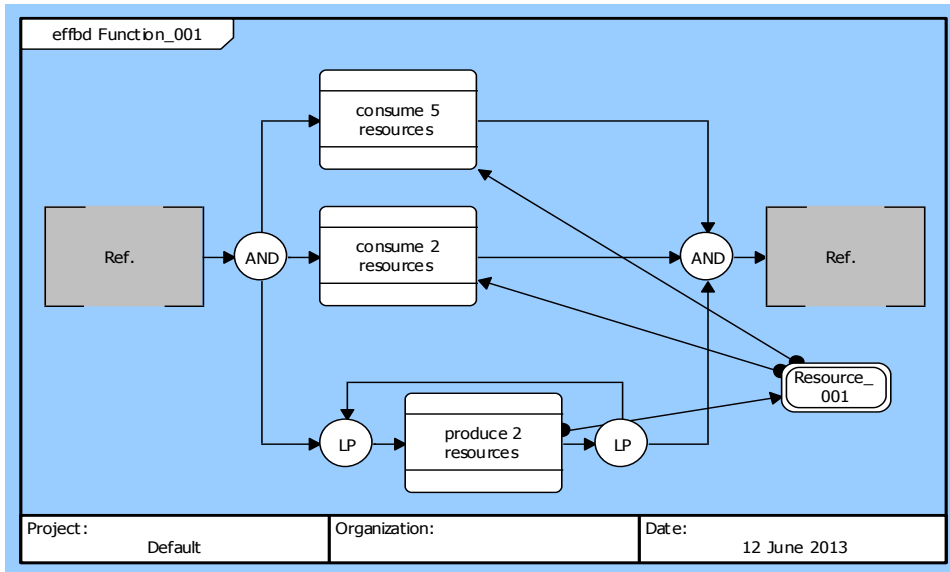
❑ בזמן ההמתנה, ייווצרו המשאבים בהדרגה וישוריונו לטובתה.

❑ הפונקציה השנייה (consume 2 resources), לא תרוץ גם כאשר יהיו 2 משאבים זמינים מאחר והם שוריינו לפונקציה הראשונה.

❑ כל עוד המשאבים משוריינים לטובת הפונקציה הראשונה, הם צבועים בצבע אדום. הם קיימים במערכת, אך לא לשימוש פונקציות אחרות, מלבד זו שממתינה להם.

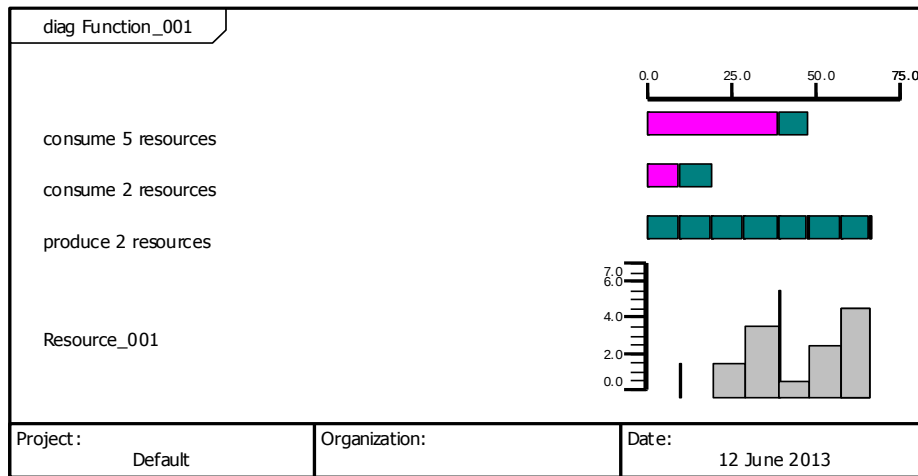
הפונקציה הצורכת 5 משאבים תקבל שליטה ראשונה לריצה, מאחר והיא נמצאת בענף העליון של AND Control.

שימוש במשאב (2)



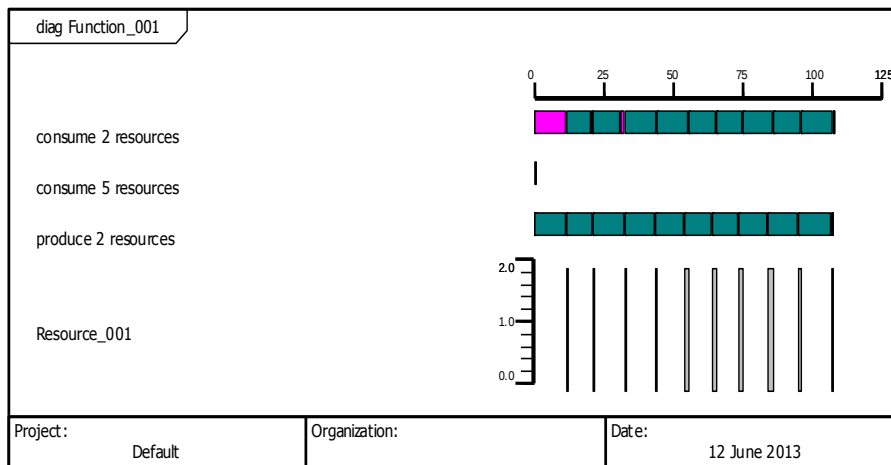
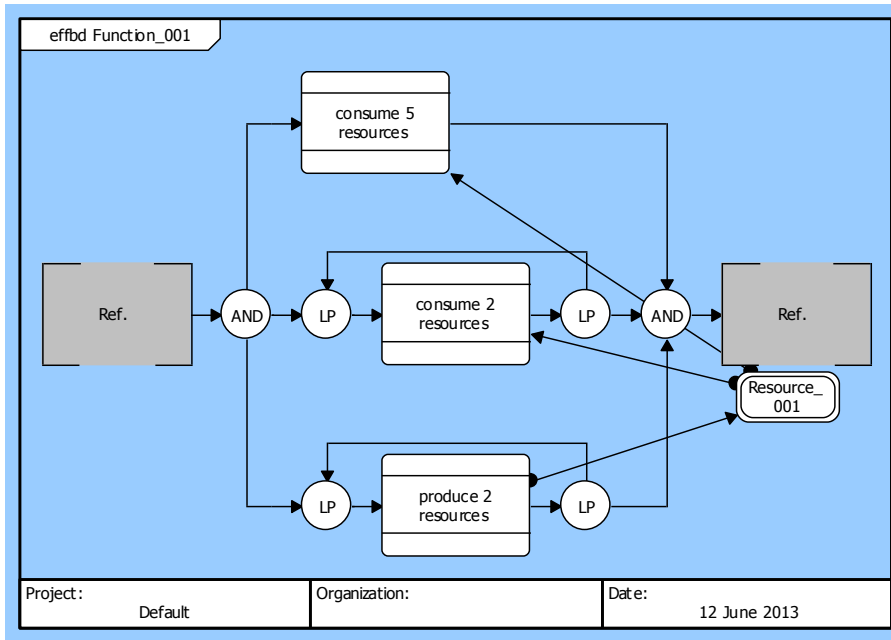
❑ הערך FALSE הוגדר לפונקציה הראשונה (consume 5 resources) – כלומר היא תמתין ל-5 משאבים זמינים ואז תרוץ.

❑ בזמן ההמתנה, היא לא "צובעת" את המשאבים לטובתה



❑ הפונקציה השנייה (consume 2 resources), תתחיל לרוץ עוד לפני שהראשונה רצה כי יהיו לה משאבים זמינים לפני הפונקציה הראשונה.

שימוש במשאב (3)



❑ הערך FLASE הוגדר לפונקציה הראשונה (consume 5 resources). – כלומר היא תמתין ל-5 משאבים זמינים ואז תרוץ.

❑ בזמן ההמתנה, היא לא "צובעת" את המשאבים לטובתה.

❑ הפונקציה השנייה (consume 2 resources), תתחיל לרוץ עוד לפני שהראשונה רצה כי יהיו לה משאבים זמינים לפני הפונקציה הראשונה. היא נכנסת ללולאה שלא נגמרת.

❑ הפונקציה הראשונה לעולם לא תיכנס לריצה, מאחר והיא "תוותר" על ריצתה לטובת פונקציה הצורכת משאבים שזמינים לה.

❑ נוצר כאן מצב של "הרעבה"

סימולציה של תהליך דינמי

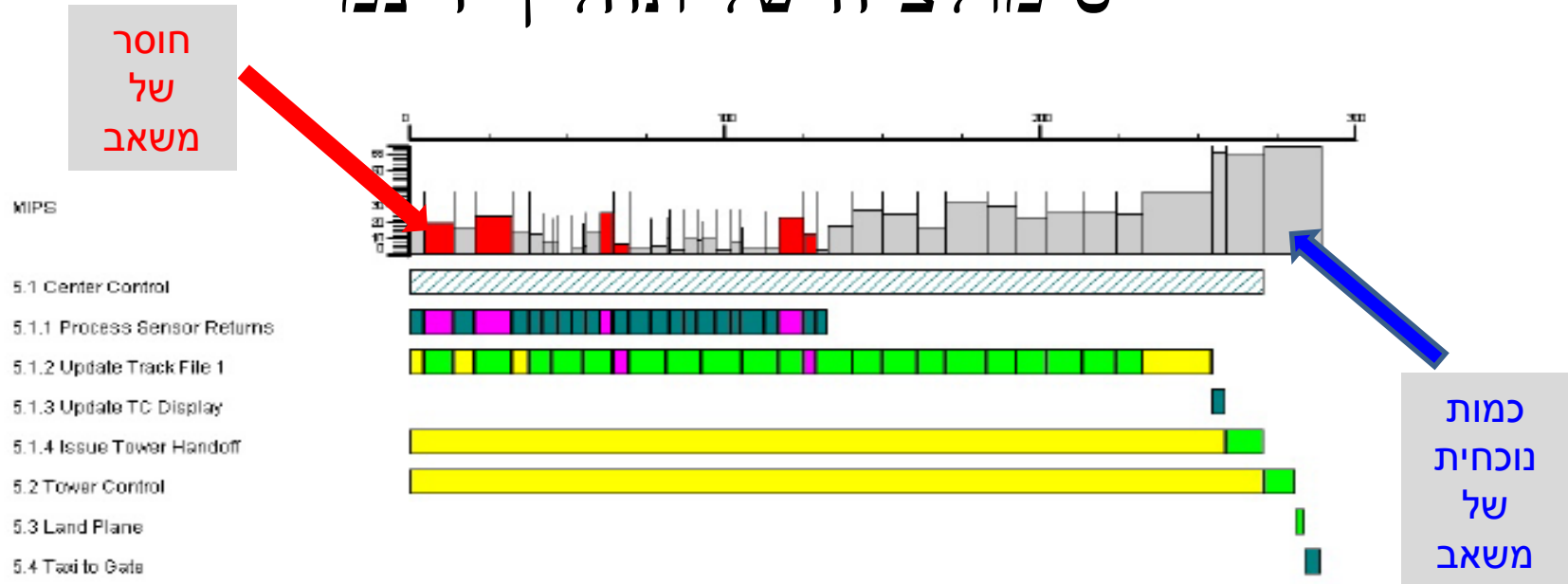
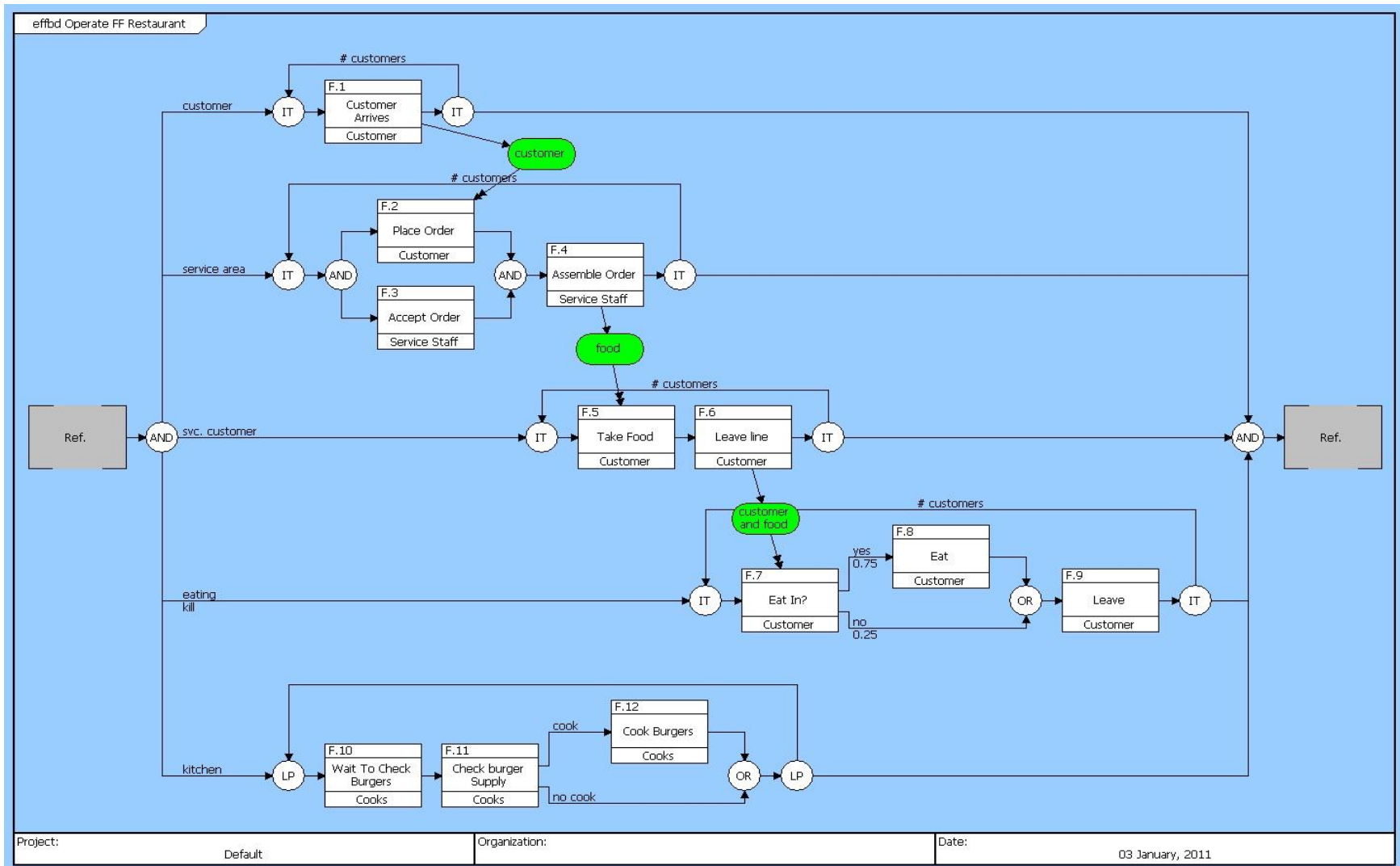


Figure 9 – Dynamic Timeline

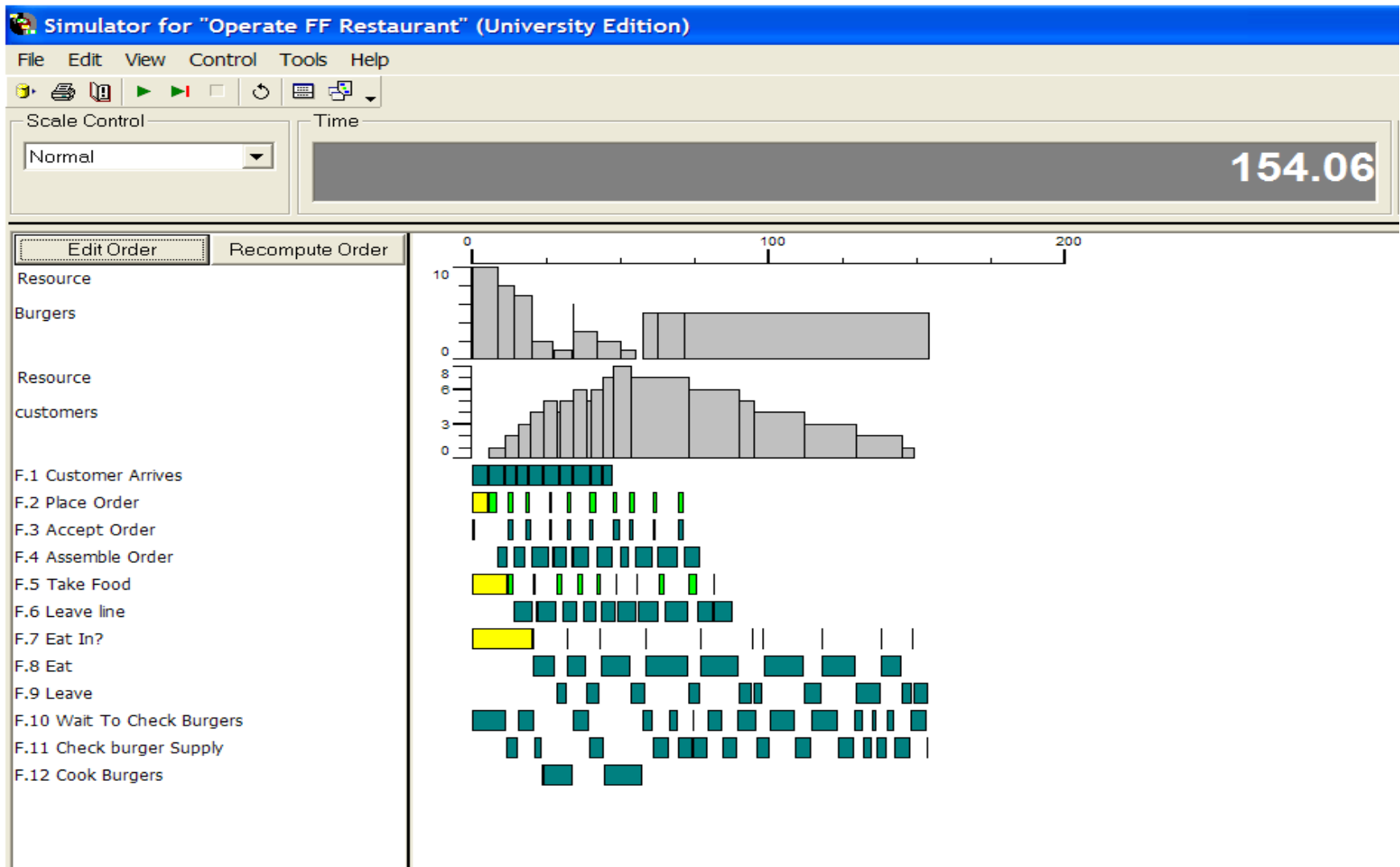
מקרא:

- צבע ירקרק-כחול (■) – הפונקציה מבוצעת ללא צורך ב"ערור" לאחר קבלת המשאבים הנדרשים.
- צבע ירוק בהיר (■) – הפונקציה מבוצעת לאחר קבלת ה"ערור".
- צבע צהוב (■) – הפונקציה מאופשרת, אך מושהית עד לקבלת ה"ערור".
- צבע ארגמן (■) – הפונקציה מאופשרת, אך מושהית עקב מחסור במשאב.
- מלבן מקווקו (▨) – פונקציות ברמה נמוכה יותר מתבצעות.

Fast Food Restaurant Example



Fast food Restaurant Simulation



Fast food Restaurant Simulation

