

קורס בדיקות תוכנה

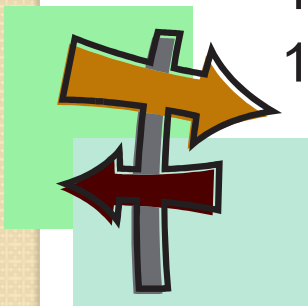
QA



פרק 3

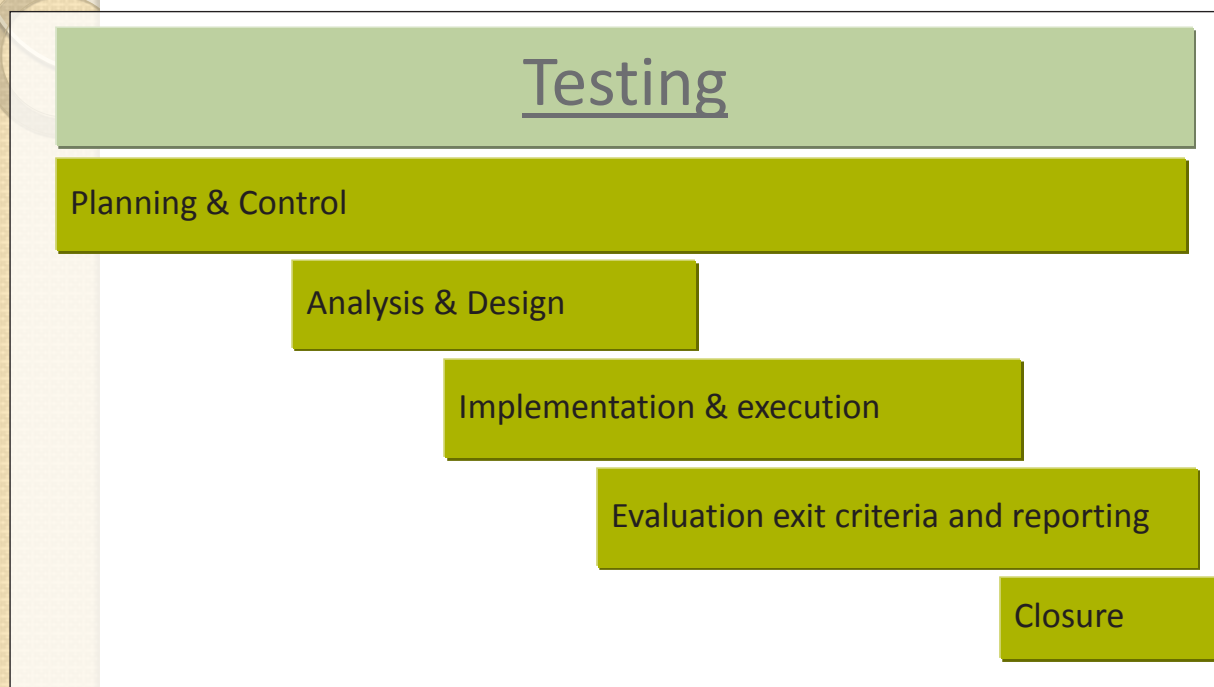
סדנא לכתיבת Software Test Plan

- 146 Test Planning – תכנון הבדיקות
- 147
 - פעילויות תכנון הבדיקות
- 151
 - תהליך תכנון הבדיקות
- 153
 - מסמך ה-STP
- 179
 - תרגיל



145

תהליך הבדיקות הבסיסי



146

תכנון פרויקט הבדיקות Test Planning

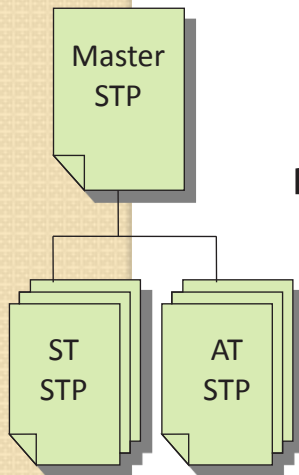
- תכנון הבדיקות נעשה עבור פרויקטי פיתוח, הטמעה ותחזוקה של מוצר

- תוצר שלב התכנון: מסמך Software Test (STP) Plan

- ניתן לפצל את ה-STP ל:

- Master STP – כולל נושאים רוחביים וכלליים של הפרויקט

- STPs ספציפיים – עבור רמות בדיקה שונות



147

תכנון פרויקט הבדיקות Test Planning

- תכנון הבדיקות מושפע ממדיניות הארגון, תכולת הבדיקות, מטרות, סיכונים, מגבלות, זמינות משאבים, ועוד

- ככל שמתקדם הפרויקט – יש יותר מידע זמין וניתן לעדכן ולפרט את התוכנית בהתאם

- מסמך ה-STP שמיש ומתעדכן במהלך חיי כל הפרויקט בעקבות משוב מתהליך הבדיקות ומידע חדש, כגון:

- שינויים בדרישות

- סיכונים חדשים

- שינוי חומרת סיכונים קיימים

- באחריות מנהל הבדיקות



148

תכנון הבדיקות: שאלות עיקריות

- מה צריך להיבדק?
- מהי גישת הבדיקות?
- מהם היחסים עם הלקוח?
- מה יהיה היקף הבדיקות האוטומטיות?
- מתי להפסיק לבדוק?
- מהם המשאבים הנדרשים?



149

פעילויות תכנון הבדיקות

- קביעת היקף הבדיקות והסיכונים
- זיהוי האובייקטים לבדיקה
- קביעת אסטרטגית הבדיקות, כולל רמות הבדיקה
- קביעת המשאבים הנדרשים
- קביעת לוחות זמנים לפעילויות הבדיקות השונות במסגרת לוחות הזמנים של הפרויקט והקבוצות השונות
- קביעת קריטריוני כניסה ויציאה
- הגדרת אחריות למשימות הבדיקות
- הגדרת מדדים ודוחות מעקב



150

תהליך תכנון הבדיקות

1. איסוף מידע מהמקורות הרלוונטיים
2. ביצוע הערכות, תכנון וכתובת מסמך ה-STP
3. קבלת משוב וקלט ממנהלי הפיתוח ושיפור ה-STP
4. ארגון ישיבת Review למסמך STP עם מנהלי הפרויקט השונים ובעלי עניין נוספים לאישור וקבלת מחויבות לביצוע התוכנית



151

מקורות מידע לתהליך התכנון

- מסמך הדרישות, כולל שינויים בדרישות ותקלות לתיקון בגרסה הנוכחית
- חוזה עם הלקוח
- מסמכי עיצוב כגון High Level design (HLD), מסמכי פונקציונאליות, ארכיטקטורה
- תכניות עבודה ולוחות זמנים של הפיתוח והפרויקט כולו
- STP קיימים
- מקורות נוספים כגון ישיבות, Mails



152

מסמך ה-STP:

◦ מבנה מסמך ה-STP:

154	הקדמה
155	תכולת הבדיקות
156	גישת הבדיקות
159	קריטריוני כניסה ויציאה
163	הערכות - Test Estimation
165	תוצרי הבדיקות
172	כוח אדם והכשרות
173	סביבות
174	תכנון זמנים
176	סיכונים
177	



153

מבנה מסמך ה-STP

- Introduction
- Features to be tested (and not to be tested)
- Customer areas to be tested (and not to be tested)
- Approach
- Entry / exit criteria
- Test deliverables
- Testing tasks
- Staffing and training needs
- Environment needs (computer resources)
- Responsibilities
- Schedule
- Risks & contingencies
- Approvals

דוגמא
STP-7

מבנה מסמך ה-STP הקדמה

- מטרת מסמך ה-STP ותכולתו
- גבולות המערכת - תיאור קצר על המערכת וגבולותיה (אילו חלקי מערכת יבדקו ואילו לא)
- מסמכים רלוונטיים – רשימת מסמכי קלט ומסמכים רלוונטיים אחרים
- מונחים והגדרות שיהיה בהם שימוש במסמך



155

מבנה מסמך ה-STP תכולת הבדיקות - Testing Scope



- מה ייבדק:
 - רשימת הפונקציונאליות החדשה שתיבדק:
 - דרישות חדשות – New requirements
 - שינויים לדרישות קודמות - Change Request (CR)
 - Customization Items
 - תקלות:
 - רשימת תקלות פתוחות מגרסאות קודמות, שיתוקנו בגרסה זו
 - בדיקות רגרסיה:
 - רשימת פונקציונאליות / תהליכים / פונקציונאליות ישנה, שלא אמורה להשתנות בגרסה זו

156

מבנה מסמך ה-STP תכולת הבדיקות – Testing Scope

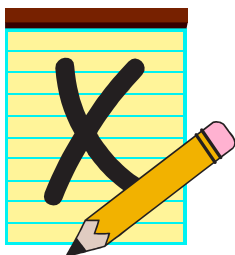
- בדיקות לא-פונקציונליות:
 - רשימת הדרישות הלא-פונקציונליות שיבדקו
- תהליכים עסקיים (Business Processes) – אופציונלי
 - רשימת תהליכים עסקיים שיבדקו



157

מבנה מסמך ה-STP תכולת הבדיקות – Testing Scope

- מה לא ייבדק:
 - ממשקים – ממשקים שבדיקתם הוגדרה כאחריות הלקוח
 - דרישות/פונקציונליות/CRs שנדחו לגרסאות עתידיות
 - ביצועים – בדיקות ביצועים שהוגדרו כאחריות הלקוח



158

הגדרת גישת הבדיקות Test Approach

- ישנן 2 סוגי גישות עיקריות:
 - הגישות המניעתיות Preventive approach – תכנון וכתובת הבדיקות מוקדם ככל האפשר בתהליך הפיתוח
 - הגישות התגובתיות Reaction approach – תכנון וכתובת הבדיקות לאחר שהמערכת קיימת ומוכנה



159

סוגי גישות

- גישות ניתוחיות – למשל בדיקות מונחות סיכונים
- גישות מבוססות מודלים – כגון בדיקות אקראיות באזורים בהם יש סטטיסטית הרבה תקלות או פעילויות עסקיות הכי נפוצות
- גישות שיטתיות – כגון בדיקות מבוססות על ניסיון, Checklists, מאפייני איכות
- גישות מבוססות תהליכים/סטנדרטים – בדיקות לעמידה בסטנדרטים מסוימים של התחום הספציפי, או שיטות Agile

160

סוגי גישות

- גישות דינמיות/חקירתיות – כגון Exploratory Testing
- גישות המבוססות על ייעוץ של מומחה – למשל לגבי מידת כיסוי הדרישות
- גישות חסכניות – שימוש חוזר בחומרי בדיקות, אוטומציה נרחבת של בדיקות רגרסיה



- ניתן לשלב כמה גישות

161

בחירת הגישה המתאימה

- בחירת הגישה מושפעת מהגורמים הבאים:
 - הסיכון לכישלון הפרויקט, סיכוני המוצר והשפעתם על אנשים, הסביבה והארגון
 - ניסיון וכישורי האנשים בטכניקות, כלים ושיטות
 - מטרת הבדיקות וחזון צוות הבדיקות
 - היבטים חוקיים – כגון תקנות פנימיות וחיצוניות בהן צריך המוצר לעמוד
 - אופי המוצר, הארגון והשוק הייעודי



162

Entry & Exit Criteria

קריטריוני כניסה ויציאה



- המטרה – להגדיר מתי נתחיל ומתי נפסיק את הרצת הבדיקות, עבור כל רמת בדיקות
- בד"כ מתבססים על נתונים מדידים:
 - מידת כיסוי הדרישות, הקוד, או סיכונים
 - מדדי Defect Density ואמינות המערכת
 - עלות
 - סיכונים שנשארו (תקלות שלא תוקנו, בדיקות שלא הורצו)
 - לוחות זמנים שאינם גמישים

163

דוגמא

Entry & Exit Criteria

Testing Level	Entry Criteria	Exit Criteria
UT	100% Requirements covered by Unit Test TCs.	100% Passed TCs 100% Executed TCs #Open Critical Defects - 3 #Open High Defects - 5 #Total Open Defects - 15
COMPONENT INTEGRATION	100% Requirements covered by COMPONENT INTEGRATION TCs.	<ul style="list-style-type: none"> 100% COMPONENT INTEGRATION TCs run 95% COMPONENT INTEGRATION TCs pass #Open Critical Defects - 2 #Open High Defects - 3 #Total open Defects – 10
ST	100% Requirements covered by ST TCs.	<ul style="list-style-type: none"> 100% ST TCs run 100% Integrative TCs run 95% ST TCs pass #Open Critical Defects - 0 #Open High Defects - 0

164

הערכות Test Estimation

- חלק מפעילויות התכנון הינו ביצוע הערכות למידת המאמץ והמשאבים הדרושים לפעילויות הבדיקות בפרויקט
- שתי גישות לביצוע הערכות:
 - גישה מבוססת מדדים
 - גישה מבוססת מומחים
- לאחר ביצוע הערכות למאמץ הנדרש - ניתן לקבוע אילו משאבים נדרשים והכנת לוחות זמנים בהתאם

165

גישה מבוססת מדדים

- הערכת המאמץ הנדרש לבדיקת הפרויקט הנוכחי בהסתמך על נתונים מפרויקטים קודמים/דומים
 - מספר תסריטי בדיקות לעומת מספר הדרישות
 - מספר ימי העבודה לכתיבת תסריטי הבדיקות
 - מספר ימי העבודה להרצת הבדיקות
 - מספר הבודקים
 - מספר התקלות שנמצאו לעומת מספר הדרישות



166

הערכות מבוססת מדדים - דוגמא

Area	Effort (Days)
New Functionality and Regression	
New Functionality	117.0
Regression	0.0
ST defects verification	12.2
Patch/service pack	
Upgrade related Activities	
Version Planning & Summary	
Effort estimation+STP+PP+Summary Report+RCA+LL	
TCM (Test Configuration management)	
Sanity execution	6.0
ST Envs. Management	

Automation Development / Maintenance and execution	
Site/UAT Support	
Send tester to site during UAT	4
UAT defects testing (post ST execution)	14.2
Other	
Review UAT TCs	0
Downtime	2.0
Other tasks required by account	
Management (Activities & People)	
PM	50.0
TL	70.0
Conversion testing	0
TOTAL (Man Days)	275.4
Training	12.7

167

גישה מבוססת מדדים

- במידה ואין נתונים מפרויקטים דומים/קודמים (למשל פרויקט של מוצר חדש) – הערכה על בסיס אומדן המקובל בתעשייה/בתחום
- סך עלות הבדיקות = 20% - 50% מסך עלות הפרויקט כולו
- כיום יש מגמה להפחית את עלות הבדיקות לכדי 12%-15% מעלות הפרויקט



168

גישה מבוססת מומחים

- הערכת המאמץ הכולל ע"י סיכום הערכת המאמץ עבור המשימות השונות
- הערכת המאמץ עבור כל משימה נעשית ע"י האדם שיבצע אותה או ע"י הערכת מומחה



169

גורמים המשפיעים על הערכות

- מאפייני המוצר:
 - איכות הדרישות ומסמכי העיצוב (DD ,HLD)
 - גודל המוצר
 - מידת מורכבות המוצר
 - דרישות לאמינות ואבטחת מידע
 - דרישות לתיעוד



170

גורמים המשפיעים על הערכות

- מאפיינים של תהליך הפיתוח:



- יציבות הארגון
- כלים בשימוש
- תהליך הבדיקות
- כישורי הבודקים
- מגבלות זמן

- תוצרי הבדיקות:

- כמות התקלות
- כמות העבודה החוזרת (Regression, Confirmation)

171

תוצרי הבדיקות Test deliverables

- רשימת תוצרי תהליך הבדיקות בכל שלב, תאריך יעד לתוצר ובאחריות מי

- לדוגמא:

- מסמך Software Test Plan (STP)
- חוקי בדיקה Test Conditions
- תסריטי בדיקות Test Cases
- דוח מצב הרצת בדיקות
- דוח תקלות
- דוח סיכום בדיקות Test Summary Report (STR)
- תסריטי בדיקות אוטומטיות

172

כוח אדם והכשרות

Staffing and training needs

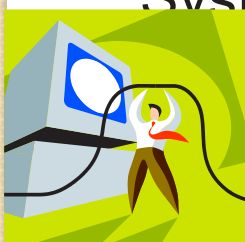
- ציין כמה עובדים נדרשים עבור כל רמת בדיקות ושלב בדיקות:
 - מספר בודקים, מפתחים (עבור UT, Integration Test)
 - ראשי צוותים (צוותי הבדיקות והפיתוח)
- הכשרות - ציין אילו הכשרות נדרשות (אם בכלל) עבור כל עובד



173

סביבות בדיקה

- הגדר את צרכי סביבות הבדיקה בהתבסס על:
 - מסמכי הארכיטקטורה וההתקנה של המערכת
 - דרישות רמות הבדיקה השונות
 - כלים נדרשים לביצוע הבדיקות
- הגדר כמות סביבות הנדרשת עבור רמות בדיקה: Integration Test ו-System Test



174

הגדרת סביבות בדיקה קריטריונים לדוגמא

- דוגמא לקריטריונים להגדרת סביבות בדיקה:
 - Hardware machines (Servers, working stations etc)
 - Architecture & Configuration
 - Disks space
 - External interface facilities
 - Input data generators
 - Utilities for results reporting
 - Special test monitor and control tools
 - licenses of 3rd party software

175

תכנון לוחות זמנים Schedule

- תכנן את לוח הזמנים עבור פעילויות הבדיקות השונות:

#	Activity	Responsible	Start Date	End Date	Comment
1	Test Plan	Test Manager			
2	Test Conditions	Testing Team Leader			
3	Test cases	Testing Team Leader			
4	Test Execution	Testing Team Leader			

176

סיכונים

- זהה את הסיכונים לפרויקט הבדיקות באזורים

הבאים:

- כוח אדם
- ידע וניסיון של צוות הבדיקות
- זמינות נתונים לבדיקות (Real Data, Synthetic data)
- סביבות הבדיקה (מוכנות בזמן, מספר הסביבות)
- כלים (האם קיימים בארגון?, רישיונות, ידע רלוונטי לאנשים)
- תוצרי תהליך הבדיקות
- לוחות זמנים ומשך הבדיקות



177

סיכונים

- מלא את הטבלה עבור כל סיכון שזוהה:

- תיאור הסיכון
- חומרת הסיכון (למשל קריטי/גבוה/בינוני/נמוך)
- השפעה על
- תוכנית טיפול במקרה שהסיכון יתממש

#	Risk	Description	Prevention	Severity	Impact on	Contingency Plan
1	HR	Not enough testers to execute the designed test cases		High	Tests execution in ST	Use the help of the developers
2						

178

תרגיל

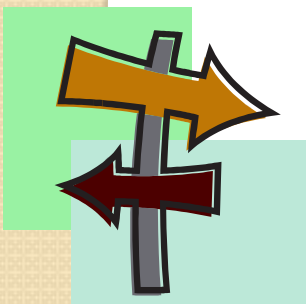
- תרגיל 3 בחוברת תרגילים: כתיבת STP
- ניתן להמציא מידע נוסף על מנת לכתוב STP מקיף
- משך התרגיל: 1 שעה



179

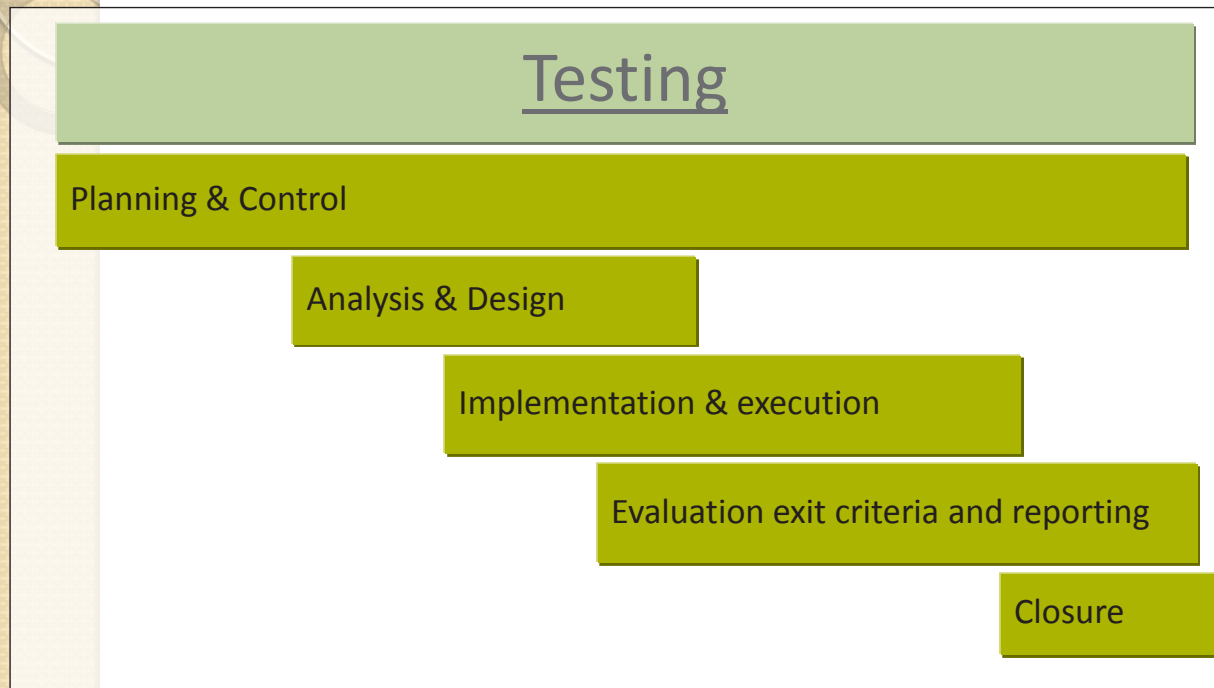
בקרה וניהול תצורה

- בקרה על תהליך הבדיקות – Test Control 182
- 183 ◦ מדדים נפוצים
- 197 ◦ פעולות מתקנות
- 198 • ניהול תצורה
- 202 ◦ ניהול תצורה ובדיקות



180

תהליך הבדיקות הבסיסי



181

בקרה על תהליך הבדיקות Test Control

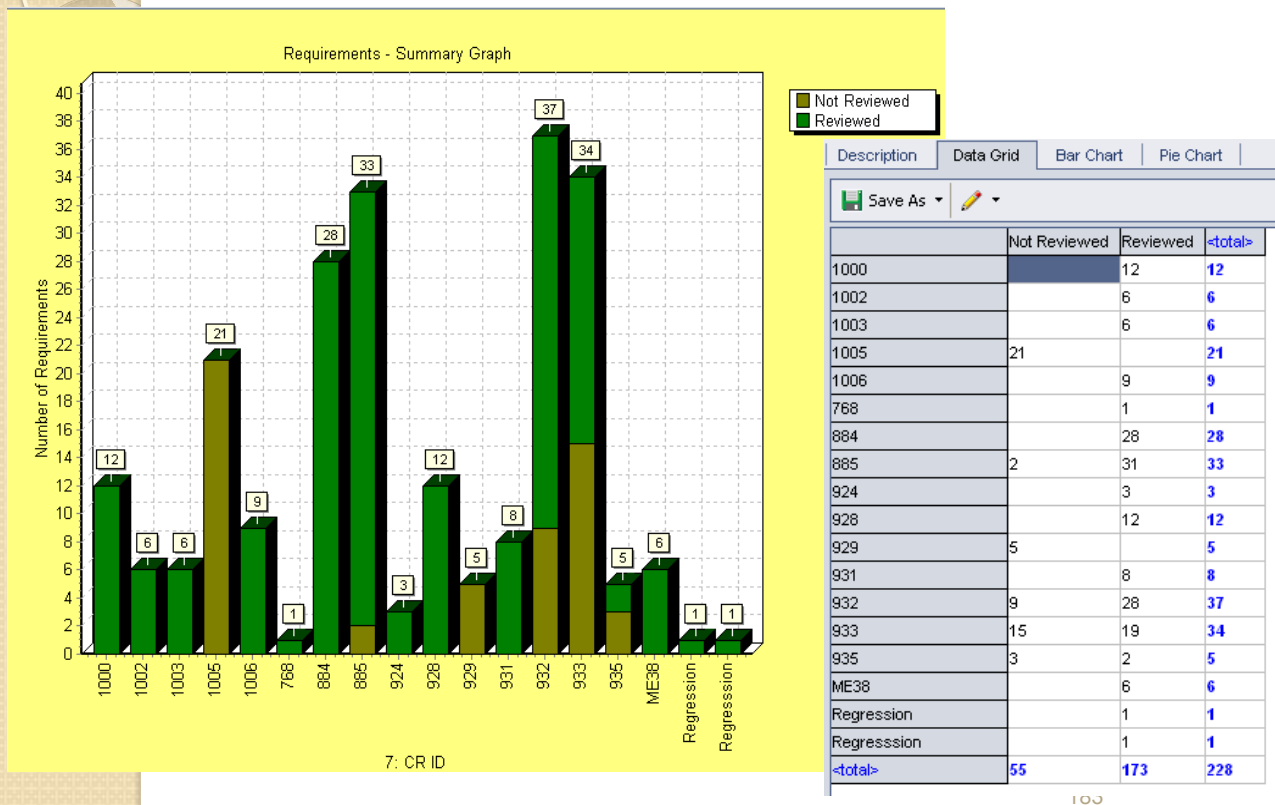
- מטרת הבקרה: לתת משוב ולשקף את מצב פעילויות הבדיקות השונות
- ניתן לאסוף את המידע בצורה ידנית או אוטומטית (בעזרת כלים)
- הבקרה מאפשרת:

control

- למדוד הגעה לקריטריון יציאה
- למדוד את מידת כיסוי הבדיקות
- להשוות את התקדמות המשימות לתוכנית העבודה או
- להשוות את מידת השימוש בתקציב לתוכניות התקציב

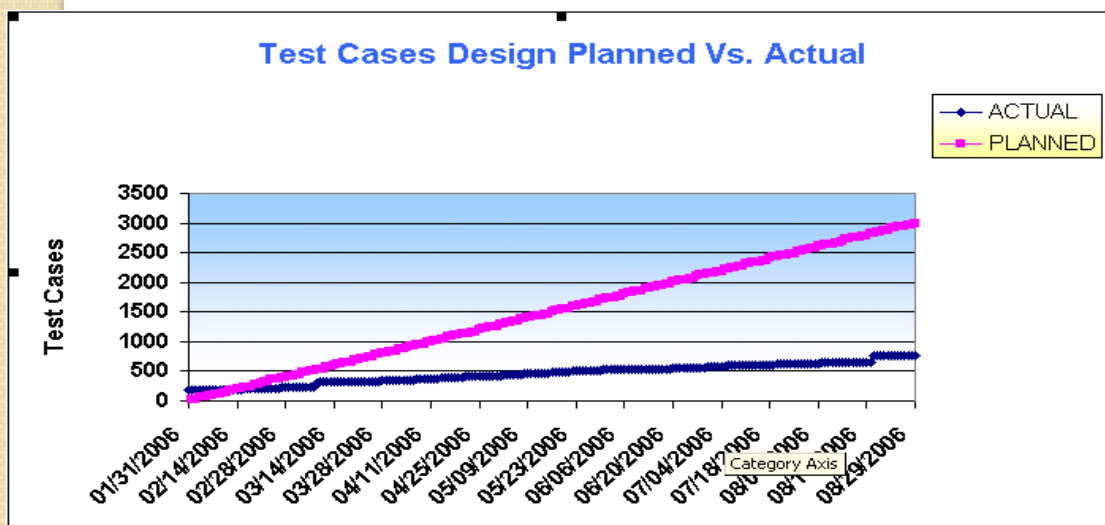
182

מדדים נפוצים: התקדמות כיסוי הדרישות



מדדים נפוצים: התקדמות הכנת Test Cases

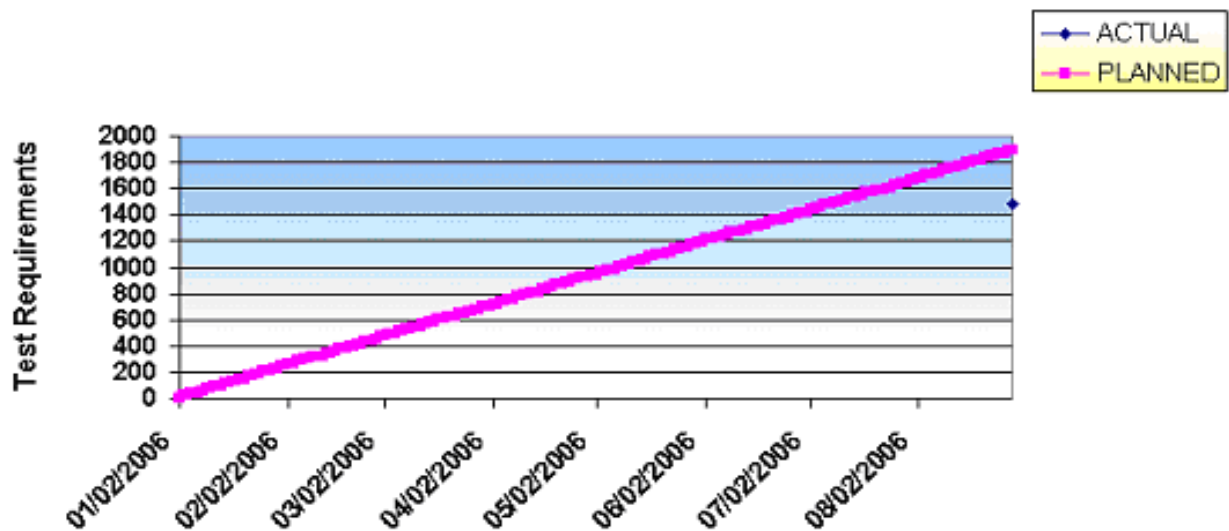
- כתיבת Test Cases – תכנון מול ביצוע לאורך זמן



מדדים נפוצים: התקדמות הכנת הבדיקות

- מידת כיסוי הדרישות ע"י TCS – בשלב כתיבת ה-TCS

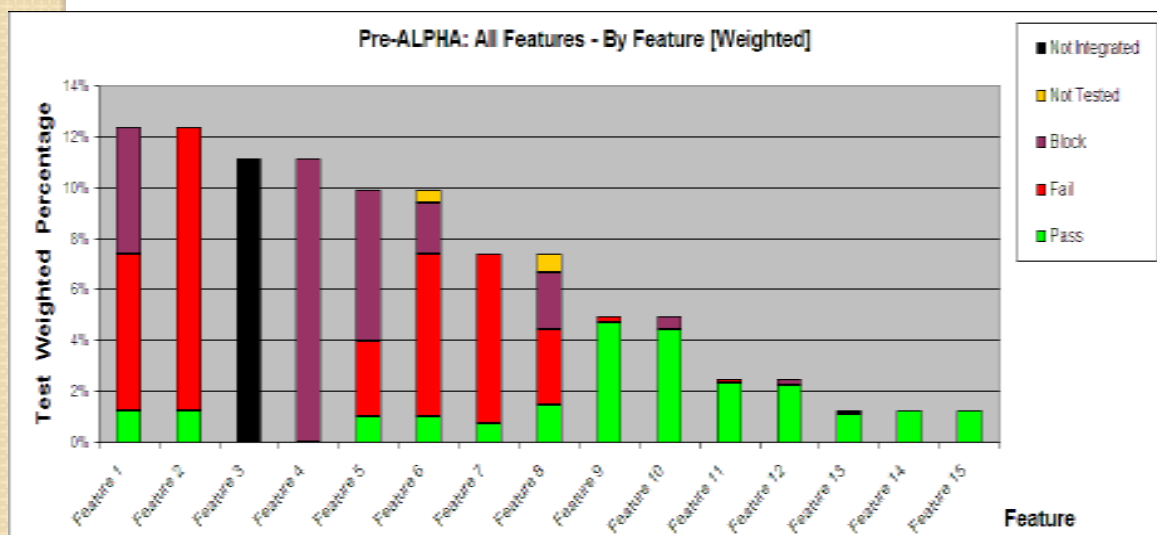
Test Requirements Coverage Planned Vs. Actual



185

מדדים נפוצים: כיסוי הדרישות ע"י הבדיקות

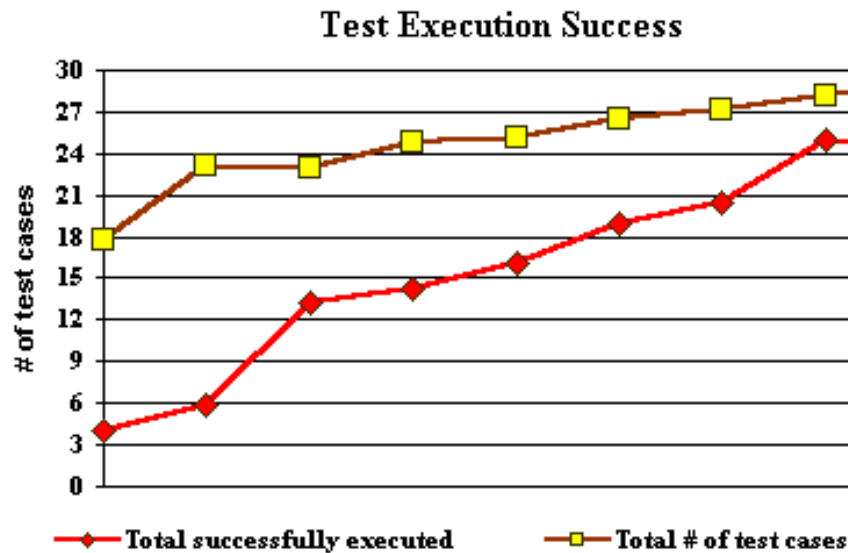
- מידת כיסוי הדרישות ע"י הבדיקות וסטטוס ההרצה עבור כל דרישה



186

מדדים נפוצים: התקדמות הרצת הבדיקות

- אחוז ה- TCs שעבר בהצלחה מתוך סך כל ה-TCs שהורצו

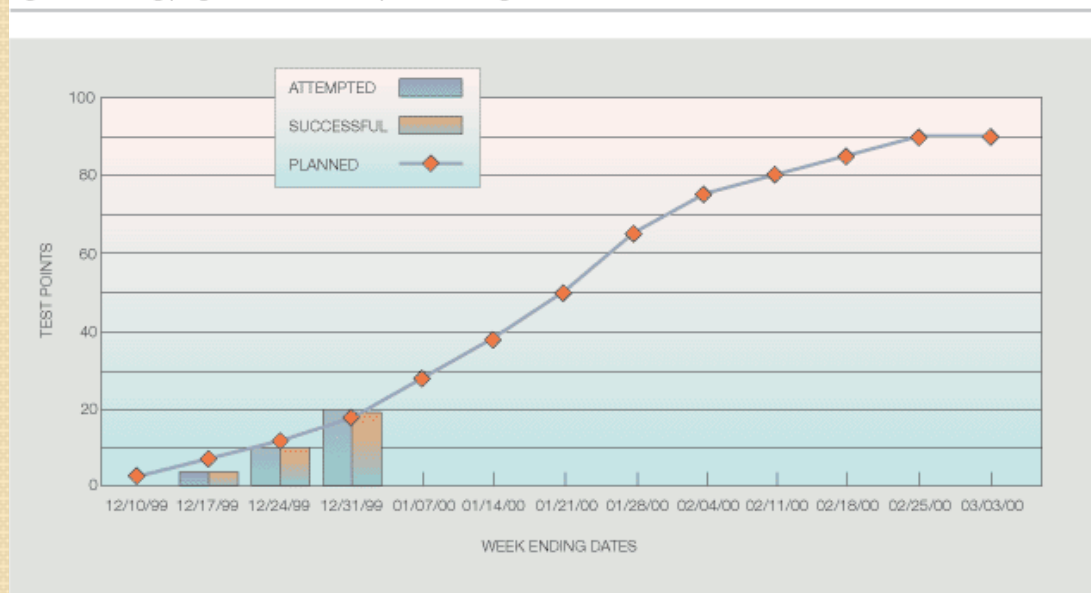


187

מדדים נפוצים: התקדמות הרצת הבדיקות

- אחוז ה- TCs שהורץ מתוך סך כל ה-TCs – תכנון מול ביצוע לאורך זמן

Figure 3 Testing progress S curve—test points tracking



188

מדדים נפוצים:

תקלות פתוחות לפי חומרה ומודול

Open Defect Statistics

Critical & High Open more than 2 days:

5

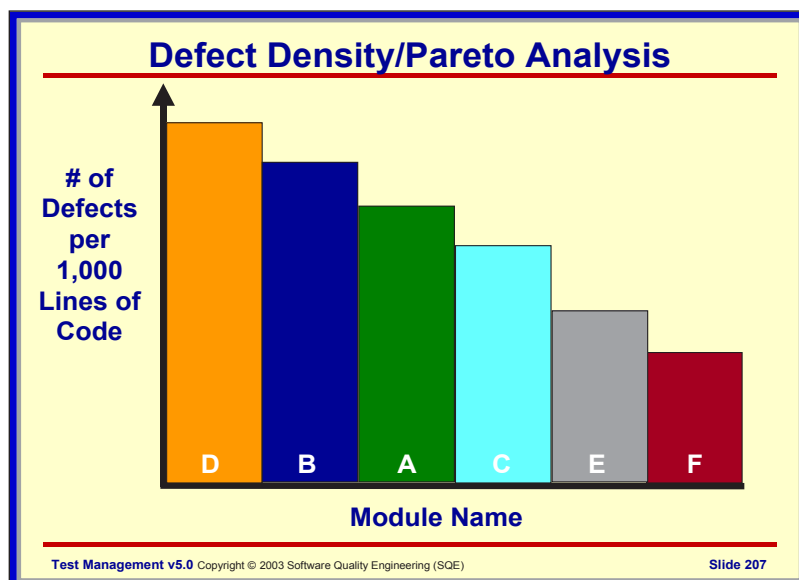
Area	1-Critical	2-High	3-Medium	4-Low	Fix	Total
CM	0	1	0	0	0	1
Infra ADBA	0	0	0	0	2	2
Infra Operational	0	1	0	0	0	1
Interfaces	0	1	0	0	0	1
OLC	0	0	0	0	0	0
PC	0	0	0	0	0	0
Rater	0	2	0	0	0	2
Ref. Table	0	0	0	0	0	0
TRB	0	0	0	0	0	0
UAMS	0	0	0	0	0	0
Totals	0	5	0	0	2	7

189

מדדים נפוצים:

Defect Density

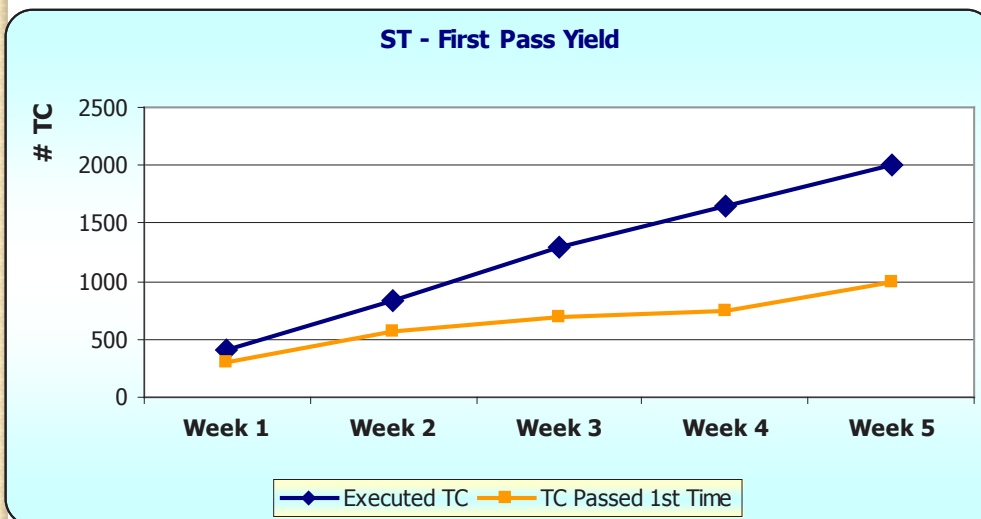
- מספר התקלות במודול/קומפוננטה מסוימת לחלק לגודל המודול / קומפוננטה (למשל LOC של המודול)



190

מדדים נפוצים: איכות הקידוד

- אחוז ה- TCs שעבר בהצלחה בריצה הראשונה מתוך סך כל ה- TCs שהורצו

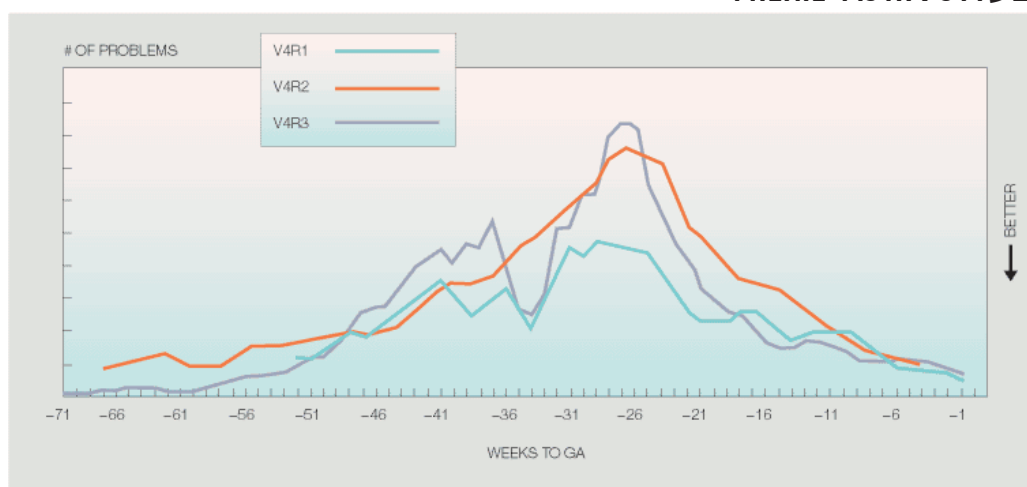


191

מדדים נפוצים: קצב גילוי תקלות לאורך זמן

השוואה בין 3 גרסאות שונות על מנת לבחון האם הגרסה הנוכחית בעלת אותה מגמה

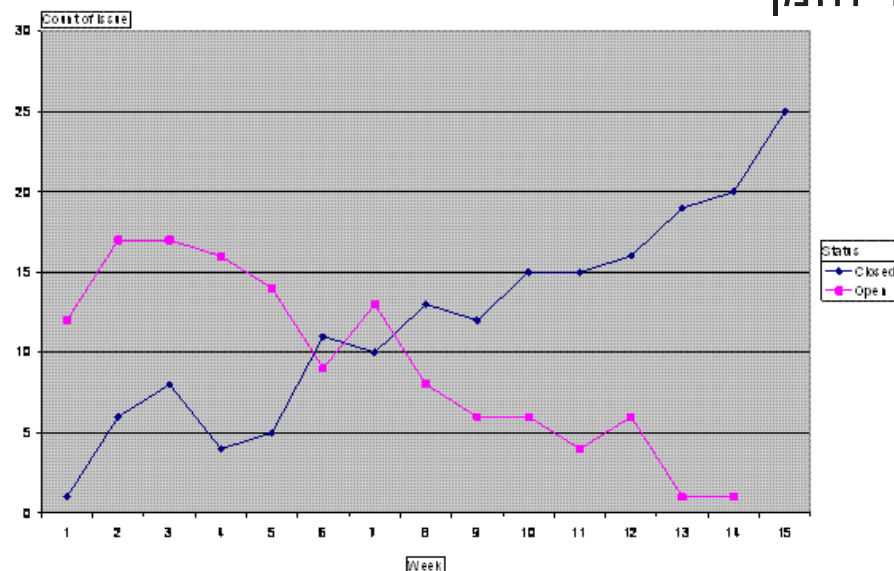
Figure 5 Testing defect arrivals metric



192

מדדים נפוצים: קצב גילוי לעומת תיקון תקלות

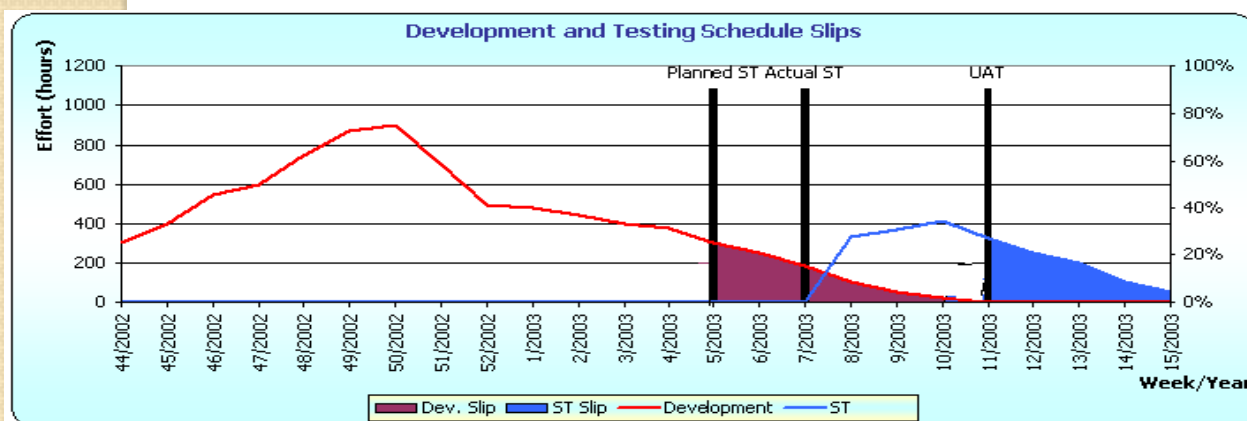
- אחוז התקלות שהתגלו לעומת אחוז התקלות שתוקנו לאורך ציר הזמן



193

מדדים נפוצים: גלישה בלוחות זמנים

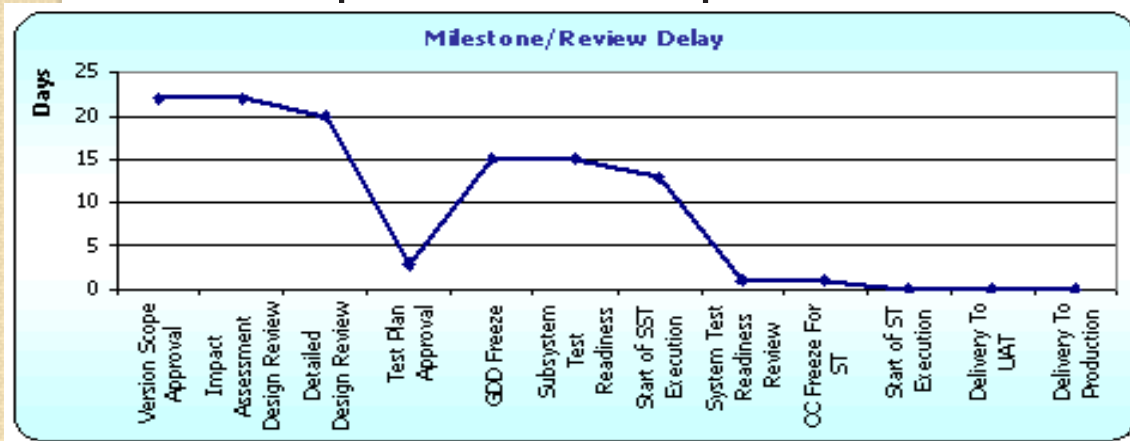
- גלישה במאמץ ותוצרי הפיתוח והבדיקות לעומת התוכנית המקורית



194

התקדמות הפרויקט – עמידה ב- Milestones

- תכנון מול ביצוע – האם ישנם עיכובים בעמידה ב- Milestones המקוריים של הפרויקט?



- חישוב: השוואה בין תכנית עבודה המקורית
Baseline לבין העדכנית ביותר

195

מדדים נפוצים נוספים

- אחוז התקדמות בהכנת סביבות הבדיקה
- תוצאות Retest (Confirmation testing)
- אחוז כיסוי הבדיקות את הסיכונים או קוד
- מידת הביטחון של הבודקים במוצר (סובייקטיבי)
- עלויות הבדיקות
- עלות התועלת של מציאת התקלה הבאה לעומת עלות תיקונה, או של הרצת הבדיקה הבאה



196

בקרת תהליך הבדיקות פעולות מתקנות

- נקיטת פעולות מתקנות בעקבות המידע הנאסף מהמדדים השונים
- פעולות מתקנות יכולות להתבצע בכל פעילות בדיקה ולהשפיע על פעילויות אחרות בתהליך הפיתוח
- פעולות מתקנות לדוגמא:
 - שינוי תעדוף הבדיקות כאשר סיכון מתממש
 - שינוי תזמון הרצת בדיקה עקב זמינות סביבת בדיקות
 - קביעת קריטריון כניסה ל-Build: תיקון תקלה חייב להיבדק קודם ע"י המפתח שתיקן אותו (Confirmation Testing)



197

ניהול תצורה Configuration Management

- ניהול ובקרת השינויים בישויות השונות של פרויקט פיתוח תוכנה, במטרה ליצור ולשמור על עקביות ביניהן
- ישויות הפרויקט להם עושים ניהול תצורה:



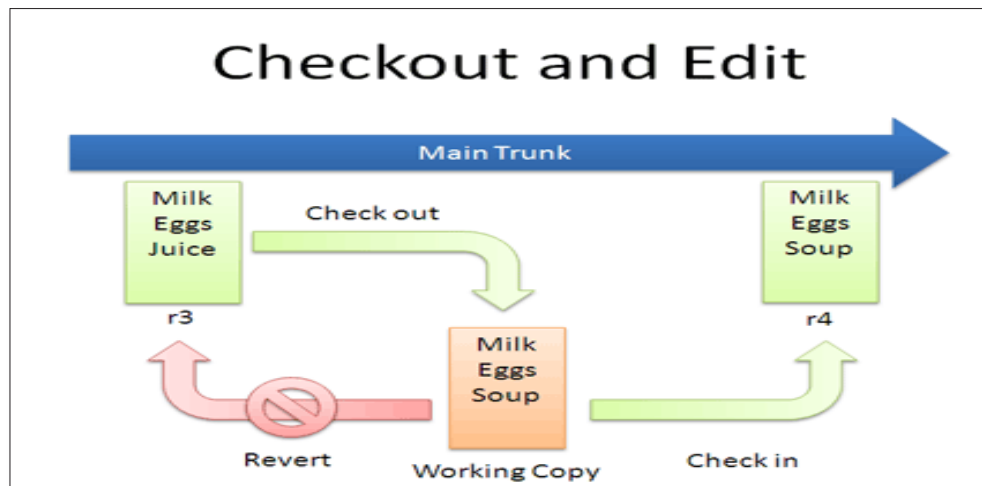
- קבצי קוד
- מסמכים
- Testware
- גרסאות של התוכנה – Builds
- נתונים
- סביבות

198

ניהול תצורה

Configuration Management

- בקרת שינויים מתבצעת באמצעות תהליך Check-out - Check-in ושמירת Baselines:



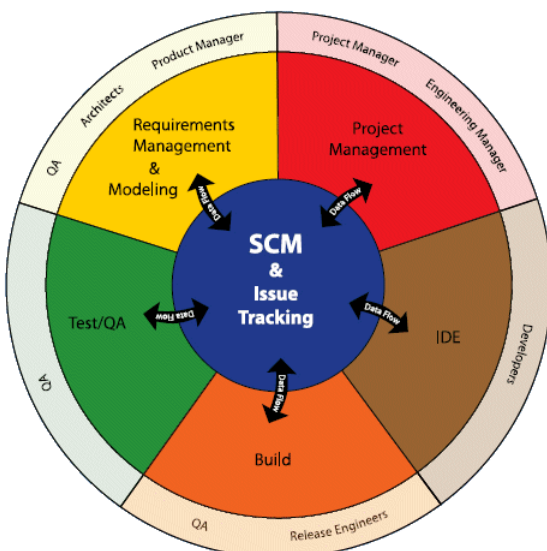
199

ניהול תצורה

Configuration Management

- ניהול התצורה מתבצע לאורך כל מחזור חיי פיתוח התוכנה

- חשוב במיוחד כאשר מפתחים את המוצר במספר בגרסאות
- ישנם כלים תומכים לניהול תצורה



200

בעיות של חוסר ניהול תצורה

- לא ניתן למצוא את הגרסה האחרונה של קובץ קוד מסוים
- תקלה שתוקנה בעבר – מופיעה שוב
- פונקציונאליות שפותחה ונבדקה – נעלמה באופן מסתורי
- תוכנית/ פונקציונאליות שנבדקה בהצלחה – לפתע לא עובדת
- הגרסה הלא-נכונה של המוצר נבדקה
- אין עקיבות בין הדרישות, המסמכים והקוד
- המפתחים עובדים על גרסה לא-נכונה של הקוד
- לא ניתן לדעת מה כוללת הגרסה שנמסרה ללקוח



201

ניהול תצורה ובדיקות

- כל פריטי הבדיקות (Testware) מזוהים
- מנוהלים על פי גרסאות
- ישנו מעקב אחר השינויים בהם
- קשורים אחד לשני ולרכיב/מערכת הנבדקת
- כל פריטי הבדיקות משויכים בצורה ברורה למסמכי המוצר הרלוונטיים ופריטי התוכנה
- כך שנשמרת עקיבות Traceability לאורך כל תהליך הבדיקות

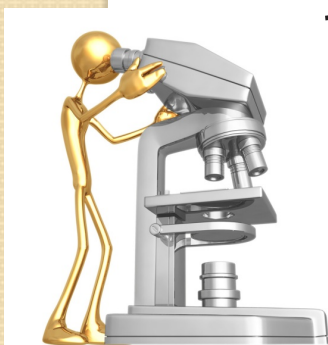


202

ניהול תצורה ובדיקות

- עבור הבדק, ניהול תצורה עוזר לזהות באופן חד-חד ערכי (ולשחזר) את הרכיב הנבדק, מסמכי הבדיקות וסביבת הבדיקות הרלוונטיים

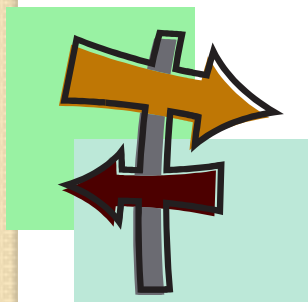
- במהלך כתיבת ה-STP יש להגדיר את תהליך ניהול התצורה, לבחור כלי תומך ולאחר מכן – להטמיע תהליך זה



203

ניהול תקלות

- מהו defect? 207
- מחזור חיים של Defect 211
- דיווח תקלות 212
- מדדי תקלות 219



204

סיפור אמיתי על תקלה

"... לאחרונה רכשתי פונטיאק חדשה ומאז שקיבלתי את הרכב, כאשר אני מביא מהחנות גלידת וניל – הרכב לא נדלק אך הוא נדלק בכל סוג גלידה אחר."

אנשי פונטיאק, שתחילה לא הבינו מה האיש רוצה מהם, החליטו לקחת את העניין ברצינות ושלחו את אחד המהנדסים להתחקות אחר הבעיה. המהנדס הצטרף למסעות הרכישה ונדהם לגלות שכל מה שנאמר היה נכון. כשהאיש רוכש גלידה בטעם וניל – המכונית מסרבת להניע בחזרה מהחנות. עם כל טעם אחר – המכונית מניעה בלי בעיות.

205

סיפור אמיתי על תקלה - המשך

המהנדס סרב להאמין שהמכונית אכן אלרגית לטעם וניל החל לרשום את כל הפרטים האפשריים: באיזה דלק השתמשו בכל פעם; איזה שעה ביום; זמן הנסיעה וכד'. במהרה התגלה פרט מעניין: כשהאיש רוכש גלידה בטעם וניל, הוא יוצא מהר יותר מהחנות. הסיבה – הוניל, שהוא הטעם הפופולארי ביותר בחנות – נמצא קרוב יותר לקופה. מכאן התגלגלו העניינים במהירות – התקלה לא נבעה מאלרגיה כלשהי לגלידה-וניל, אלא מאדי קיטור שנכלאים במערכת הזנת הדלק. לוקח להם זמן מסוים להשתחרר – פחות ממה שנדרש כדי לרכוש גלידת פיסטוק – ויותר ממה שצריך בשביל גלידה-וניל.

"גם בעיות שנראות מטורפות לגמרי, עשויות להיות אמיתיות, ופתרון נמצא רק כשחושבים עליהם בראש פתוח. העיקר – הגישה".

206

מהו Defect?

- במהלך הרצת בדיקה - כל תוצאה שאינה תואמת לתוצאה הצפויה הינה תקלה (Defect, Incident)!
- כל defect צריך להיות מתועד
- יש לעקוב אחר התקדמות הטיפול בכל Defect מרגע פתיחתו ועד לסגירתו ווידוא נכונות התיקון
- על מנת לעקוב בצורה יעילה אחר כל ה- Defects יש להגדיר ולהטמיע תהליך וחוקי מיון



207

מתי מגלים Defects?

- ניתן למצוא ולדווח תקלות במהלך:
 - כתיבת הקוד
 - תהליך ה-Review על מסמכים
 - הרצת הבדיקות
 - כאשר המערכת ב-Production
- סוגי Defects:
 - תקלות קוד
 - תקלות סביבה
 - בעיה במסמכים (מסמכי דרישות, עיצוב HLD, DD, מסמכי בדיקות, מסמכי התקנה, מדריך משתמש, HELP, וכו')
 - בעיה במערכת העובדת (Production)

208

מטרת דיווח ה-Defect

- לספק מידע מדויק על התקלה, כיצד לשחזרה ומידע רלוונטי נוסף (כגון צילומי מסך, קבצי log) על מנת לאפשר למפתח הרלוונטי לזהות, לבודד ולתקן את התקלה
- לתת למנהלי הבדיקות אפשרות לעקוב אחר איכות התוכנה והתקדמות הבדיקות
- לספק רעיונות לשיפור תהליך הבדיקות

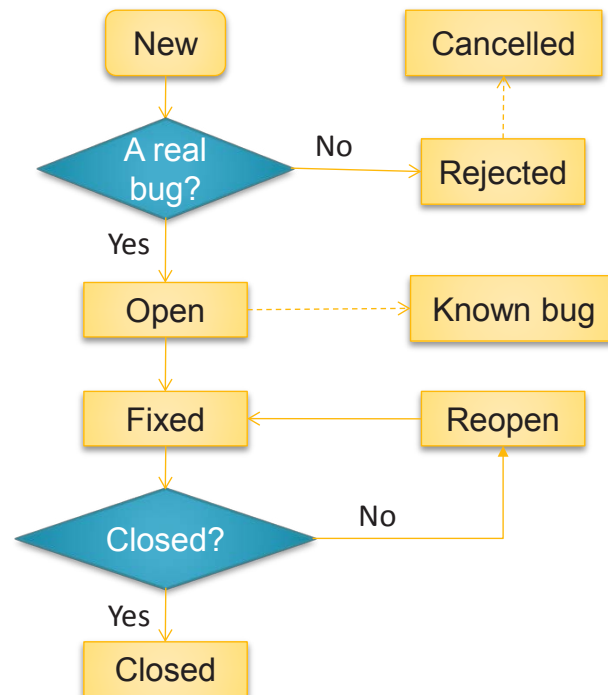


209

דוגמא: Defect form report Quality Canter

210

מחזור חיים של Defect דוגמא



211

Incident report items IEEE 1044 / 829

- Identification
- Title
- Problem description
- How to reproduce
- Actual outcome
- Expected outcome
- Reproducibility
- Test case
- Date
- Status
- Class (defect, design)
- Severity
- Priority
- Module / function
- Version
- Environment
- Tester
- Test level
- Attachments
- Change history

חומרה Severity

- מידת החומרה Severity של תקלה נקבעת על פי ההשפעה שלה על המערכת ותפקודה
- סולם לקביעת רמת החומרה (על פי Rex Black):
 1. איבוד מידע, נזק לחומרה או בעיה בטיחותית
 2. פונקציונאליות מרכזית אינה עובדת ואין Workaround
 3. פונקציונאליות מרכזית אינה עובדת ויש Workaround
 4. פונקציונאליות עובדת חלקית
 5. תקלה קוסמטית, זניחה

213

עדיפות התיקון Priority

- מידת העדיפות שיש לתת לטיפול בתקלה זו לעומת הטיפול בתקלות אחרות
- ה-Priority משקף את נקודת מבט הלקוח ו/או הבדיקות:
 - עד כמה חשוב ללקוח/לבדיקות לקבל את התיקון כמה שיותר מהר
- מנהל הפיתוח יכול לשנות את דרגת ה-Priority, על פי שיקולי כוח אדם, עדיפויות וסיכונים

214

סולם דוגמא Priority

Priority level	Description	Response time
1 – Very high (immediate)	Must be fixed as soon as possible due to critical severity and/or feature importance to the customer	Defect should be responded to within 24 hours
2 – High (Next release)	Must be fixed in the next planned release. Usually will reflect high severity defects or risk to the project timeline	Within the next release
3 – Medium (final release)	Should be fixed before the final delivery (to AT or to production)	No later than the final release
4 – Minor / low	Can be ignored or fixed in a future release/ Such bugs should be documented as 'known defects' as part of the release notes	

215

הדגשים בדיווח תקלות

• Summary:

- חשוב שיתאר בשורה אחת מה בדיוק הבעיה, כך שגם אנשים שאינם רואים את הפירוט יבינו מה התקלה
- לא לכתוב דברים כלליים או עמומים, כגון:
 - המערכת קורסת
 - לא ניתן להגדיר לקוח חדש
 - Error בתשלומים
 - דוח לא נכון

216

הדגשים בדיווח תקלות



• Description:

- חשוב לכתוב בצורה עניינית, להיצמד לעובדות
- לא לכתוב סיפורים ארוכים
- תוצאה בפועל ותוצאה צפויה
- לתאר מה הפעולות שנעשו לפני הופעת התקלה, מה היה מצב המערכת
- הוראות איך לשחזר את התקלה
- הוסף כל פרט שיכול לעזור למפתח לבודד את התקלה, כולל צילומי מסך, קבצי log ומסמכים אחרים
- פרטים מזהים על הגרסה, הסביבה, תסריט הבדיקה
- היה דיפלומטי

217

תרגיל

- תרגיל 4 - דיווח תקלות בחוברת תרגילים
- 10 דקות



218

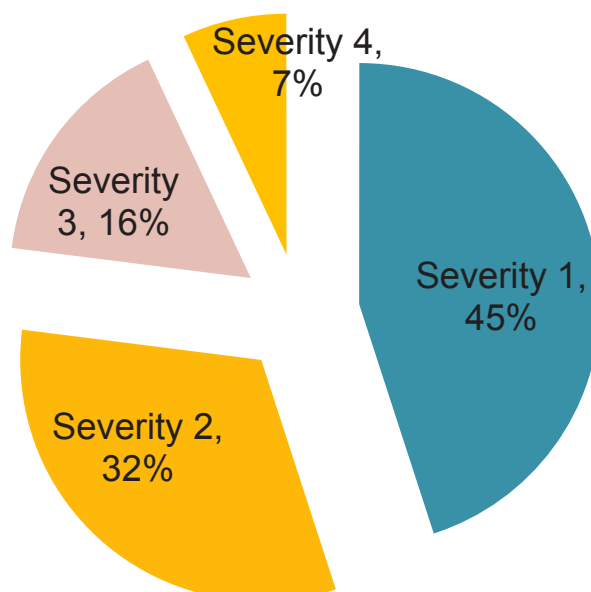
מדדי תקלות

- מדדים עוזרים לנו לאסוף מידע מדיד לגבי תהליכי הפיתוח והבדיקות, להעריך את יעילותם ולשפרם
- חלק מהמדדים מופקים באופן תקופתי (למשל שבועי) וחלקם בסוף גרסה



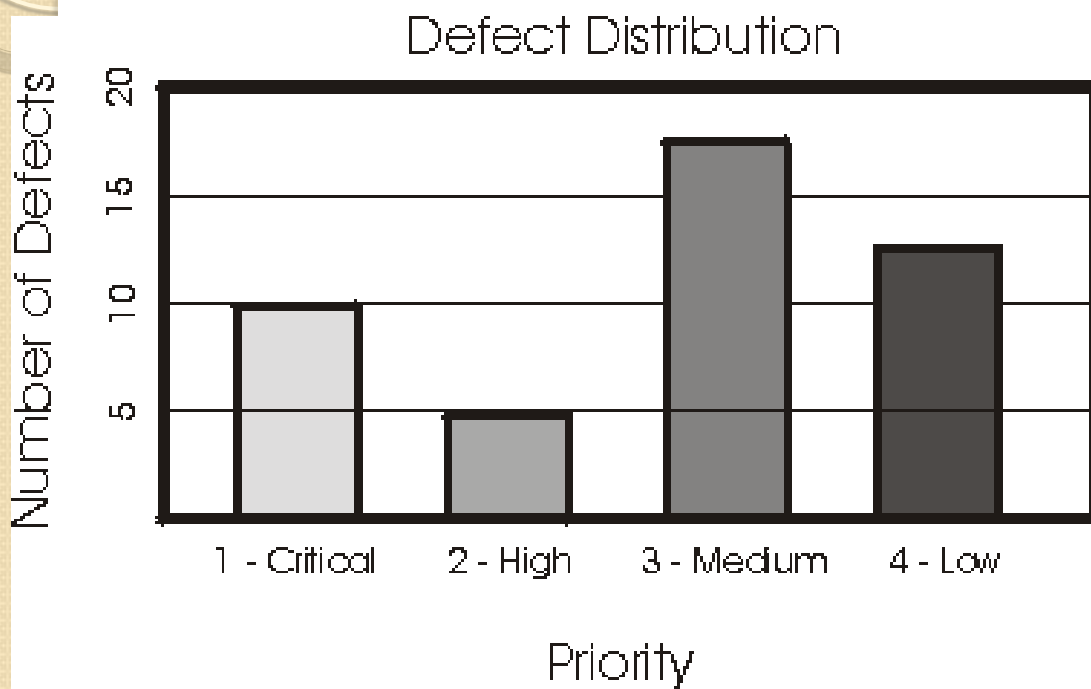
219

דוגמא למדדי תקלות: Defects by severity



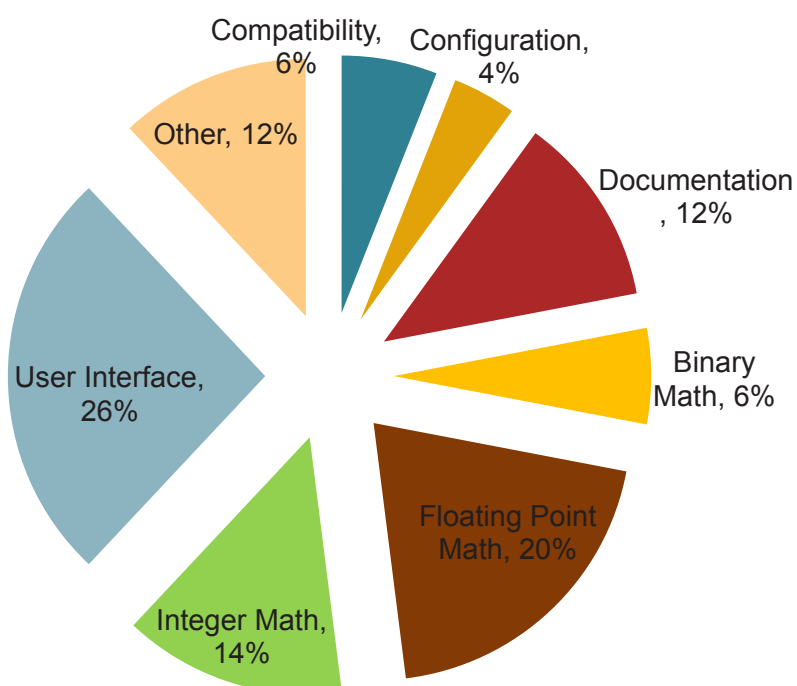
220

דוגמא למדדי תקלות: Defect distribution to priority



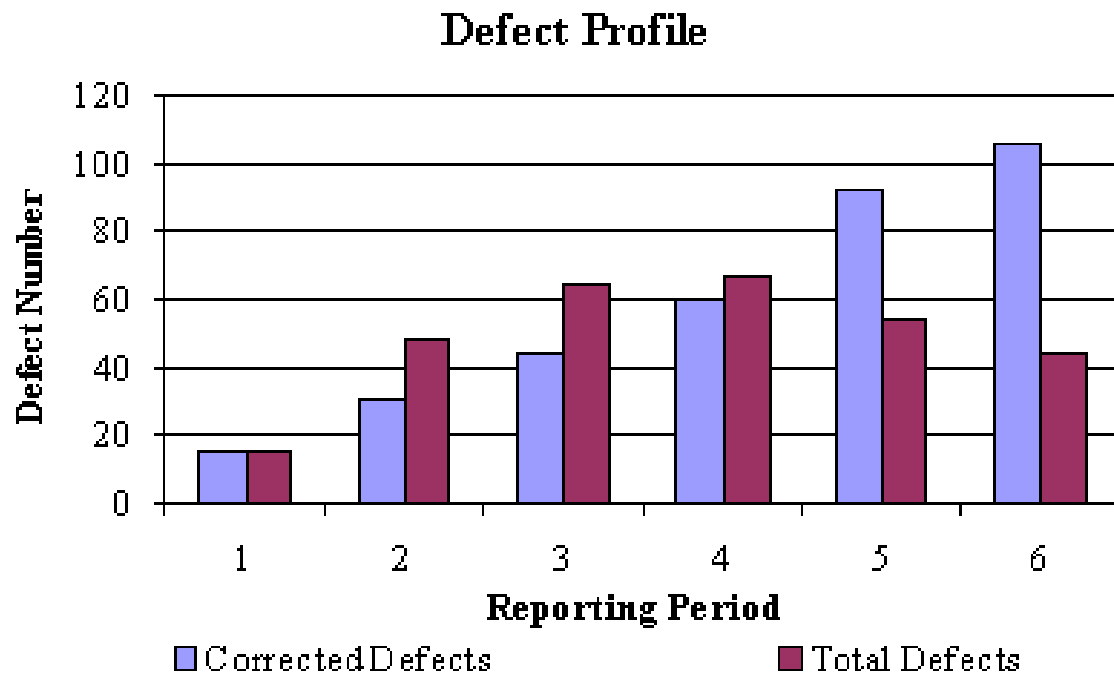
221

דוגמא למדדי תקלות: Defects by Software Area



222

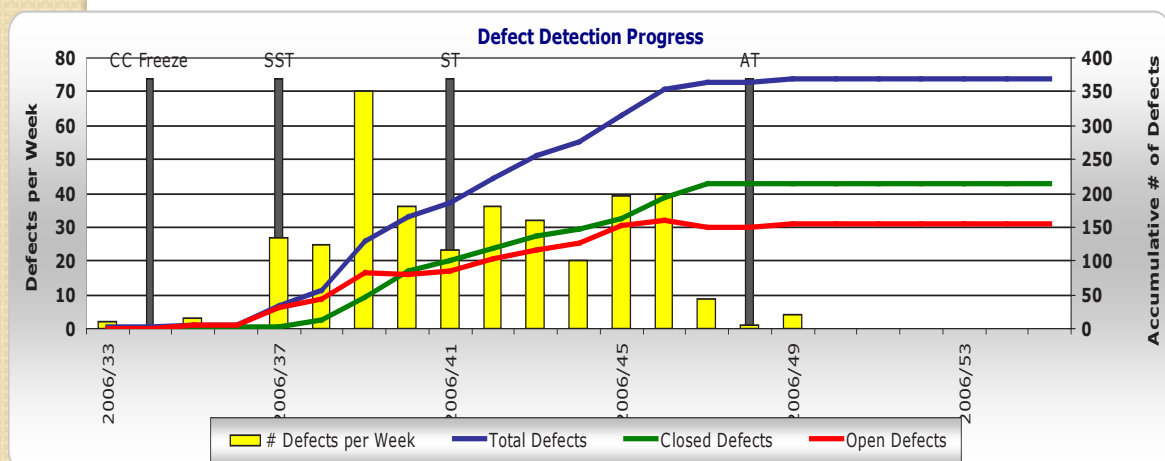
דוגמא למדדי תקלות: Reported defects vs. corrected defects over time



223

דוגמא למדדי תקלות: קצב גילוי התקלות לפי רמת בדיקה

- כמות התקלות שהתגלו בכל שבוע וברמות הבדיקה השונות: בדיקות אינטגרציה, בדיקות מערכת



224

דוגמא למדדי תקלות: Defect status and aging reports

Defect Dashboard - ST For XXXX

Defects List

Author: winrunner

Defect ID Not (11870 or 11892)
Detected by Group On Site Testing' Or 'System Test'
Detected in Version/Build 7.2

Defect ID	Detected in Version/Build	Status	Detected By
11687	RBP 7.2	Closed	einatw
11702	RBP 7.2	Cancelled	deepakta
11703	RBP 7.2	Closed	einatw
11705	RBP 7.2	Closed	alonas
11711	RBP 7.2	Cancelled	alonas
11719	RBP 7.2	Closed	einatw
11720	RBP 7.2	Cancelled	alonas
11751	RBP 7.2	Closed	einatw
11759	RBP 7.2	Closed	iditf
11764	RBP 7.2	Closed	einatw
11791	RBP 7.2	Closed	alonas
11860	RBP 7.2	Closed	shmuelam
11861	RBP 7.2	Closed	veredpe
11862	RBP 7.2	Closed	veredpe

Defect Dashboard - ST For XXXX

Defects Aging

Author: winrunner

Defect ID Not (11870 or 11892)
Detected by Group On Site Testing' Or 'System Test'
Detected in Version/Build 7.2

	On Testing	
	07-System Test	Total
0 Day		0
1 Day		0
2 Days		0
3 Days		0
4 Days		0
5 Days		0
6 Days		0
1 Week	1	1
2 Weeks		0
3 Weeks		0
1 Month		0
2 Months		0
More...	2	2
Total	3	3

דוגמא למדדי תקלות: אחוז התקלות המבוטלות

Cancelled Defects

