קורס בדיקות תוכנה

QA



פרק 1

מבוא לבדיקות תוכנה

• הצורך בבדיקות תוכנה

• מה הן בדיקות תוכנה?

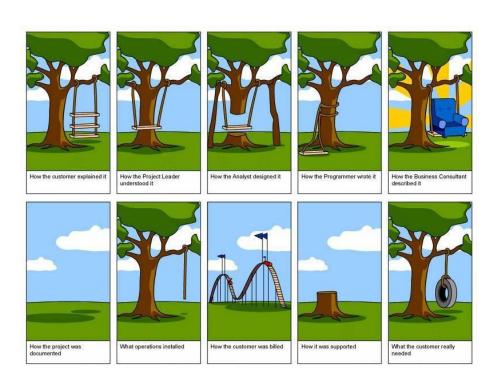
• בדיקות ואיכות

• עקרונות מנחים בבדיקות תוכנה

30 PISTQB מה זה •



הצורך בבדיקות תוכנה



הצורך בבדיקות תוכנה

- מערכות תוכנה הן חלק בלתי נפרד מחיי השגרה בכל
 פן בחיים
 - בנקים, מכוניות, תקשורת, מכשירים רפואיים, מכשור
 משרדי, מכשור צבאי, ועוד
 - שגיאה במערכת תוכנה כאשר תהליכים לא מסתיימים כפי שציפינו
- שגיאות שכאלה גורמות לעיתים נזק כלכלי רב, איבוד
 זמן ואף פציעה או במקרים מסוימים, אף מוות

הצורך בבדיקות תוכנה

- תקלות תוכנה עולות למשק האמריקאי 60 מיליארד דולר בשנה !!
- אפשר היה לחסוך יותר משליש מסכום זה ע"יביצוע בדיקות תוכנה טובות יותר

(National Institute of Standards & Technology, "The Economic Impacts of Inadequate Infrastructure for Software Testing", May 2002)



ATM



בדיקות תוכנה – סיפורי אימה

ב- 1999 התרסק לווין מזג אויר שנשלח למאדים עקב תקלת תוכנה "טיפשית" – חישוב יחידות מידה מוטעה (ס"מ ואינצ'ים)

ב-1988 הפילה ספינת טילים אמריקאית (USS Vincennes) שהייתה במפרץ הפרסי מטוס נוסעים איראני מכיוון שזיהתה אותו כ- F14 עוין. 290 נוסעי המטוס נהרגו

ב-1992 תקלה בתוכנת השיגור של שירות אמבולנסים לונדוני גרמה למוות של כמה אנשים מכיוון שהאמבולנסים לא הגיעו מיידית, כפי שהיו אמורים להגיע.

בין 1985-1987 נגרמו כמה מקרי מוות של חולי סרטן עקב חשיפתם לכמות קרינה גבוהה מידי במכשיר Therac-25 כתוצאה מתקלת תוכנה

גורמים לתקלות תוכנה

כשל – **Failure**

תקלת תוכנה – Defect, Bug







- טעות אנוש

Mistake,

- בביצוע פעולה
- ביצוע פעולה שלא הייתה אמורה להתבצע
- סטייה מהתוצאה הצפויה
- בקוד התוכנה
 - במערכת
- במסמכים הקשורים אליה

דוגמאות

- :Mistake, Error טעות אנוש
- הגדרה חלקית של דרישה, דרישה ש"שכחו", איפיון חלקי
 של דרישה
- "ילדים עד גיל 5 לא ישלמו כרטיס, ילדים עד גיל 18 ישלמו 10 ש"ח לכרטיס " – האם ילד בן 5 משלם או לא?
 - :Defect, Bug תקלת תוכנה
 - וf X<=5 Then... :בקוד המפתח כתב ∘ במקום If X<5 then
 - :Failure כשל
 - הבודק מריץ טסט בו נבדק ערך גיל 5 ומקבל סה"כ
 לתשלום: 0 ש"ח, במקום 10 ש"ח

גורמים לתקלות תוכנה

תקלות בתוכנה, מערכות או מסמכים עלולות ליצור
 כשלון, אך לא כל תקלה תיצור כשלון בסופו של
 דבר



גורמים לתקלות תוכנה

- גורמים הקשורים בבני אדם:
 - חוסר ניסיון ∘
 - ∘ הנחות בסיס שגויות
 - ∘ לחץ זמן ∘
- ∘ מורכבות המערכת: קוד ותשתית, ריבוי ממשקים
 - ∘ טכנולוגיה: ישנה מדי/ חדשה מדי
 - גורמים סביבתיים:
- קרינה, מגנטיות, שדות חשמליים, זיהום אוויר:
 יכולים להשפיע על החומרה, המשפיעה על ביצועי
 התוכנה

תפקיד הבדיקות

- ביצוע בדיקות מחמירות עשוי להפחית מהסיכון להופעת תקלות במהלך עבודת המערכת
- ביצוע איכותי של בדיקות בשלב הפיתוח תורם להעלאת איכות התוכנה, במידה והתקלות שהתגלו מתוקנות לפני שחרור המוצר
- ווידוא עמידת המוצר בדרישות חוזיות, חוקיות
 וסטנדרטים ספציפיים נעשה ע"י בדיקות

בדיקות תוכנה – מהן?

- בדיקות תוכנה כל בדיקה של רכיב תוכנה
 או מערכת תוכנה, במטרה לוודא עמידה
 בדרישות המערכת
 - יש להגדיר את תנאי הבדיקה, סביבת הבדיקות, והתוצאות צפויות לפני הרצת הבדיקות



בדיקות תוכנה – מהן?

- תהליך הבדיקות כולל פעילויות לפני ואחרי ביצוע
 הבדיקות עצמן וכולל בדיקות דינאמיות וסטטיות
- בדיקות דינמיות (<u>Dynamic Testing</u>) : בדיקות אשר במהלכן מריצים את הקוד של הרכיב/המע<mark>רכת</mark> הנבדקת
 - **בדיקות סטטיות** (<u>Static Testing</u>): בדיקות רכיב/מערכת מבלי להריץ את הקוד של המערכ<mark>ת</mark>
 - מעבר על מסמכי אפיון וקוד של המערכת (<u>Review</u>) ◦
 - (Static Analysis Tools) ניתוח סטטי באמצעות כלים ◦

בדיקות תוכנה – מהן?

- תכנון וכתיבת תסריטי בדיקות בהסתמך על מסמכי המערכת (<u>Test basis</u>) בשלב מוקדם בתהליך הפיתוח תורם למניעת תקלות מלהיכנס לקוד
- מעבר על מסמכי אפיון של התוכנה תורם אף
 הוא במניעת תקלות מלהיכנס לקוד

מטרות הבדיקות

- מציאת תקלות •
- מתן בטחון ומידע לגבי איכות התוכנה
 - מניעת תקלות •



מטרות בדיקות ורמות בדיקה

- קיימות מטרות בדיקה שונות עבור רמות בדיקה
 שונות:
- ∘ **בבדיקות Fact**ory Acceptance Test) **FAT** המט<mark>רה:</mark>
 - להכשיל את המערכת כמה שיותר על מנת למצוא את מירב התקלות ולתקנן לפני שחרור המוצר
 - ∘ בבדיקות Site Acceptance Test) **SAT** בבדיקות ∘
 - לוודא עבודה תקינה (לפי הדרישות) של המערכת וקבלת בט<mark>חון •</mark> במערכת

מטרות בדיקות ורמות בדיקה

- ∘ **בבדיקות תחזוקה** (Maintenance Testing) המטרה:
- לוודא שאין תקלות חדשות כתוצאה משינויים שהוכנסו למוצר 🔹
 - בבדיקות תפעול (Operational Testing) המטרה:
 - להעריך את אמינות המערכת או רמת זמינותה •



Testing ISTQB הגדרה של

The process consisting of all life cycle activities, both static and dynamic, concerned with planning, preparation and evaluation of software products and related work products to determine that they:

satisfy specified requirements,
to demonstrate that they are fit for purpose



כמה לבדוק?

and to detect defects

- לא ניתן לבצע בדיקות ממצות של כל הצירופים
 האפשריים להפעלת התוכנה (testing)
- ההחלטה מתי להפסיק את הבדיקות צריכה לקחת בחשבון:
 - ∘ חומרת הסיכונים (ברמת המוצר והפרויקט)
 - ∘ מגבלות של זמן ותקציב
- שהוגדרו מראש (Exit Criteria) עמידה בקריטריוני איכות ∘
 - ההחלטה נלקחת על ידי בעלי העניין בפרויקט,בהסתמך על מידע שנאסף מהבדיקות



בדיקות ו-Debugging

• בדיקות ו-Debugging הינם דברים שונים:

בד<u>יקות</u>

מראות כישלונות המערכת הנגרמים מתקלות, אך אינם מראות את המקור לתקלה (את קטע הקוד הפגום), נעשות ע"י בודקים

Debugging

ניתוח התקלה ומציאת מקורה (בקוד), תיקונה ובדיקת התיקון ברמת Unit Test, נעשה ע"י המפתחים

Re-Testing

בדיקת תיקון התקלה ע"מ לוודא שהוא מסיר את הכשל, נעשה ע"י בודקים

בדיקות ואיכות

- בעזרת הבדיקות ניתן למדוד את איכות
 התוכנה באמצעות מספר התקלות שהתגלו
 - תהליך הבדיקות בוחן את איכות התוכנה:
 - ∘ מבחינת הפונקציונאליות והמאפיינים שלה
- ∘ מבחינת היבטים בלתי פונקציונאליים (אמינות, שימושיות, תאימות, ביצועים, אבטחת מידע, ועוד)
- תהליך בדיקות מתוכנן כהלכה מוריד את רמת
 הסיכון בשימוש בתוכנה

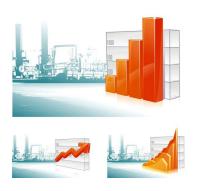
בדיקות ואיכות



- ∘ ככל שמתוקנות יותר תקלות שהתגלו בבדיקות
 - ∘ ככל שיותר בדיקות רצות בהצלחה
- אם מספר התקלות שמתגלות הוא נמוך או אפסי ∘
 - אבטחת איכות ניתוח מקור התקלותבפרויקטים קודמים משמש:
 - לשיפור תהליך הפיתוח ∘
- למניעת חלק מהתקלות ולשיפור איכות המערכת העתידית

בדיקות ואיכות

בדיקות הן אחת מפעילויות אבטחת האיכות,
 לצד פיתוח והטמעת סטנדרטים לפיתוח,
 הדרכות וניתוח תקלות





אבטחת איכות ובקרת איכות

Quality Assurance (QA) אבטחת איכות

Quality Control (QC)=
Testing
בקרת איכות = בדיקות

אבטחת איכות = בקרה על תהליכים

- שם כולל לכלל הפעולות הננקטות על מנת לייצר מוצר ברמת איכות נדרשת
 - שימוש בתהליכים מוגדרים ומתועדים של
 פיתוח תוכנה
 - בכל אחד משלבי מחזור החיים של פיתוח התוכנה ננקטים אמצעים שונים ומגוונים להבטחת האיכות של המוצר הסופי:
- ∘ נהלים, תקנים, הדרכות, בקרת איכות, בדיקות, מדדים, Reviews, מבדקים פנימיים

בקרת איכות תוכנה = בקרה על המוצר

- תהליך שנועד לבדוק תאימות של המוצר אל
 מול הדרישות
- המטרה של בדיקות תוכנה היא לוודא
 שהמוצר עונה על הדרישות שהוצבו (ולידציה)
 - בקרת איכות כוללת:
 - Reviews, Static Analysis : בדיקות סטטיות
 - Unit Test, Integration Test, בדיקות דינמיות: ∘ System Test, Acceptance Test
 - סוגי בדיקות: פונקציונאליות ולא-פונקציונאליות ∘

עקרונות מנח<mark>ים</mark> בבדיקות תוכ<mark>נה</mark>

- 1. בדיקות מצביעות על נוכחות תקלות, אך אינן יכולות להוכיח כי אין תקלות
- בדיקות מורידות את ההסתברות להמצאות תקלות שלא נמצאו בתוכנה, אך גם אם לא מתגלות תקלו<mark>ת,</mark> אין הדבר מצביע על נכונות התוכנה
 - 2. בדיקות ממצות אינן אפשריות
 - בדיקת כל האפשרויות לערכי קלט ותנאי קדם אינהאפשרית במגבלות הפרויקט
- שימוש בניתוח סיכונים ותעדוף עוזרים במיקוד מאמצי
 הבדיקות באזורים הבעייתיים

עקרונות מנחים בבדיקות תוכנה

3. בדיקות מוקדמות:

- פעילויות הבדיקות צריכות להתחיל מוקדם ככל
 שניתן בתהליך פיתוח התוכנה וצריכות להתמקד
 במטרות שהוגדרו מראש
 - .4 אשכול תקלות:
- ∞ מספר קטן של מודולים מכיל את רוב התקלות



עקרונות מנחים בבדיקות תוכנה

- 5. פרדוקס ההדברה:
- אם נריץ אותה קבוצת תסריטי בדיקות שוב ושוב− היא לא תמצא תקלות חדשות לאחר זמן
- יש לשנות את תסריטי הבדיקות באופן קבוע על מנת למצוא תקלות חדשות
 - 6. בדיקות הן תלויות הקשר:
- בדיקות עבור מערכת קריטית (בטיחות/בריאות) יהיו שונות מבדיקות עבור אתר לממכר אלקטרוני



עקרונות מנחים בבדיקות תוכנה

- .7 אשליית היעדר התקלות:
- מציאת תקלות ותיקונן לא תעזור אם התוכנה לא עומדת בצרכי המשתמש ובציפיותיו



מה זה ISTQB?

- International Software) ISTQB ארגון דוסוס (Testing Qualification Board 2002 באדינבורו, סקוטלנד ב- 2002 <u>http://www.istqb.org</u>
 - הארגון הינו בינלאומי, ומסמיך למקצועות
 בדיקות תוכנה בעולם המחשבים
 - ארגון ללא מטרות רווח •
- מפרסם תוכניות לימודים עבור כל הסמכה ומילון מונחים מקצועיים בתחום (Glossary)

ISTQB

- הסמכות אפשריות:
- ∘ Foundation Level הסמכה לבודק תוכנה, רמה בסיסית
- ∘ Advanced Level הסמכות בנושאים מתקדמים, ניהוליים
 - Expert Level •
 - 41 מדינות חברות (מרץ 09)
 - (08 מעל 100,000 מוסמכים (אוקטובר •



- ISTQB מדינות

- · UK
- Netherland
- Germany
- France
- Belgium
- Sweden
- Finland
- Denmark
- Norway
- Japan
- New Zealand
- Nigeria

- Spain
- Portugal
- Austria
- Switzerland
- Bulgaria
- Romania
- Serbia
- Montenegro
- Czech Republic
- Slovakia
- Brazil
- Israel

- Poland
- Russia
- Ukraine
- Turkey
- India
- China
- Bangladesh
- Malaysia
- Korea
- Australia
- America
- Canada

? ITCB מה זה

- Israeli Testing Certification Board = ITCB
 - השלוחה הישראלית של ה-ISTQB
 - ארגון ללא מטרות רווח
 - מטרותיו:
 - ∘ לקדם את מקצוע בדיקות התוכנה כמקצוע בישראל
 - ∘ להסמיך אנשי בדיקות בהסמכות ISTQB השונות
 - ∘ לתרגם את מבחני ההסמכות לעברית



על הקורס

- הקורס מבוסס על הסילבוס של STQB ברמת Foundation ברמת
 - :מונחים על פי
- י מילון המונחים של STQB Glossary − ISTQB ∘ 2010
 - BS (British Standard) 7925-1 תקן •



References

- Software Testing in the real world / Edward Kit
- A Practitioner's Guide to Software Test Design / Lee Copland
- Software Testing Foundations / Andreas Spillner, Tilo Linz, Hans Schaefer



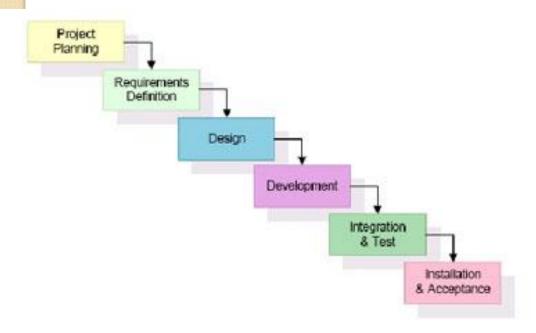
מודלים בפיתוח תוכנה

• מודלים לפיתוח תוכנה

37	∘ מודל מפל המים
39	∨ מודל ה-V ∘
42	∘ המודל הספירלי
45	Agile Model •
61	תהליך הבדיקות הבסיסי
64	חרגיל

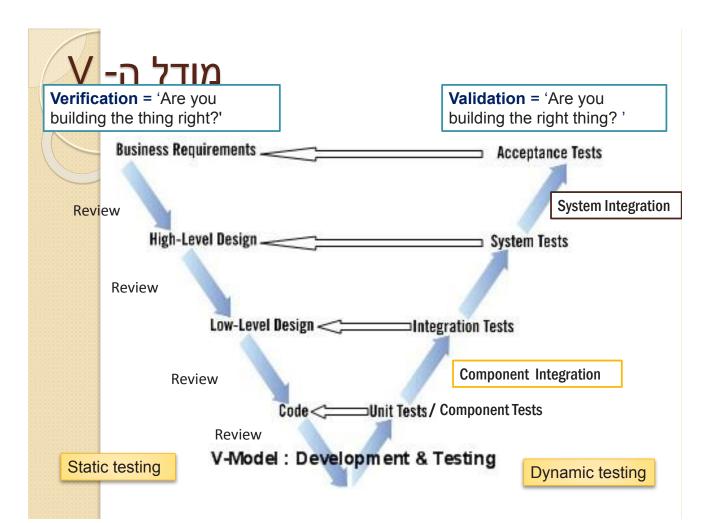


מודל מפל המים Waterfall



מודל מפל המים

- פיתוח התוכנה מתבצע בתהליך שיטתי ולוגי המורכב
 משלבים מוגדרים־היטב שאין לפסוח עליהם
- השלבים מבוצעים בטור, אחד אחרי השני, ובכל שלב יש
 מיקוד במשימה עיקרית אחת בלבד
 - איסוף וניתוח כל הדרישות כולן קודם לתחילת הפיתוח
 - הבדיקות נערכות בסוף התהליך, לפני מסירה ללקוח
- יזמן פיתוח מוצר ארוך ביותר, משלב הגדרת הדרישות וע<mark>ד •</mark> לקבלת משוב מהלקוח



ע-מודל ה-

- רמות הבדיקה:
- Unit Test / Component Test
 - Integration Test •
 - Component Integration ·
 - System Integration
 - System Test •
 - Acceptance Test •
- מטרות בדיקה שונות עבור כל רמת בדיקות

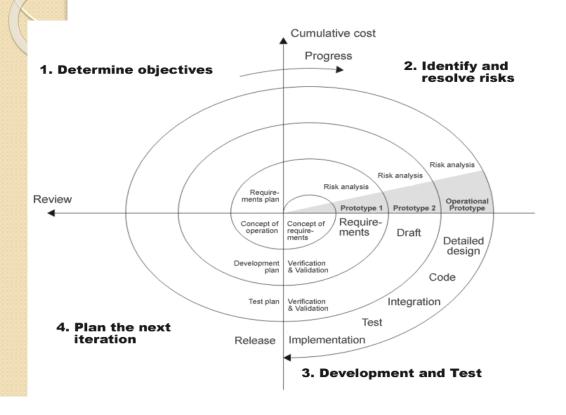
Verification & Validation אישור ווידוא

- Verification אישור תהליך אבטחת איכות שמטרתו לאשר כל תוצר בתהליך הפיתוח שהוא שלם, נכון וטוב
 - של תוצרי הפיתוח בכל שלב Review כעשה ע"י •

'Are you building the thing right?'

- Validation ווידוא- פעולת ההוכחה שתוכנה מסוימת מבצעת בדיוק את מה שהוגדר במפרט שלה
- נעשה ע"י הרצת בדיקות על המוצר ובדיקת התנהגותו •
 'Are you building the **right**thing?'

המודל הספירלי / איטרטיבי Iterative / Spiral development model



המודל הספירלי

- פיתוח התוכנה הינו תהליך מחזורי או ספירלי
- כל דרישות המוצר מחולקות לגרסאות קטנותיותר
- כל גרסה כוללת את כל שלבי פיתוח התוכנה
 לפי מודל ה-V או מפל המים
- הקטנת הסיכון בפיתוח המוצר: קבלת משוב מהלקוח בשלב מוקדם יותר ולא בסוף פיתוח המוצר כולו
 - אימות ווידוא מוקדמים עבור כל שלב בכל
 גרסה

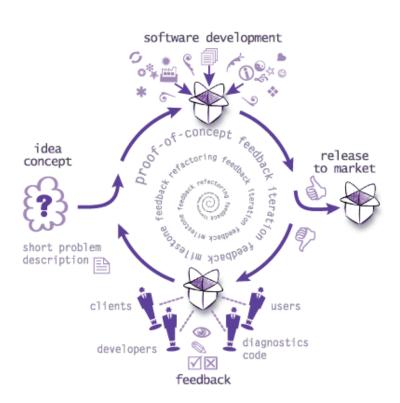
המודל הספירלי

- גמיש לשינויים שינויים בדרישות במהלך תהליך הפיתוח יכולים להיכנס לגרסה הבאה, ללא זעזועים לגרסה הנוכחית
 - חשיבות גבוהה לבדיקות רגרסיה
 - שימוש חוזר בתסריטי בדיקה שפותחו
 בגרסאות קודמות = חסכון במאמץ, עלות
 וזמנים
 - יעיל במיוחד בפרויקטי פיתוח של מוצרים גדולים ומורכבים

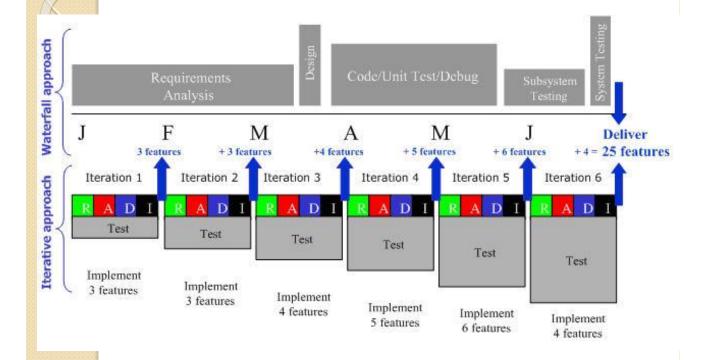
Agile Software מה זה ?Development

- Agile Software Development פיתוח תוכנה זביז
 היא גישה המניחה שפיתוח תוכנה הוא ביסודו בעיה
 אמפירית, ולא ניתן לפתור אותה בשיטות המתבססות על
 חיזוי או תכנון
- גישה זו מניחה שלא ניתן להגדיר במלואה תוכנה מסוימת קודם לפיתוחה בפועל, ומתמקדת במקום זאת בשיפור יכולתו של הצוות לספק תוצרים במהירות ולהגיב לדרישות העולות תוך כדי הפיתוח
- גישה זו היא הרחבה של הגישה האיטרטיבית, והיא שמה
 דגש רב על יכולת התגובה לשינוי, יעילות ואיכות

Agile SW Development



פיתוח תוכנה באיטרציות



Agile Development -עקרונות ה

- שביעות רצון לקוח ע"י מסירה מהירה ועקבית של
 תוכנה יעילה
 - תוכנה עובדת נמסרת לעיתים תכופות (תוך שבועות במקום חודשים)
- תוכנה עובדת היא הבסיס למעקב אחר התקדמותהפרויקט
 - שינויים מתקבלים בברכה (אפילו שינויים מאוחרים)
 - שיתוף פעולה צמוד על בסיס יומי בין אנשי ה-Business למפתחים

Agile Development -עקרונות ה

- תקשורת ישירה (פנים אל פנים) היא סוג התקשורתהטוב ביותר
 - עובדים בעלי מוטיבציה ואמינים הם הלב של הפרויקט
 - תשומת לב תמידית לאיכות העיצוב (Design) ומצוינות טכנית
 - פשטות •
 - צוותים המנהלים את עצמם
 - הסתגלות תמידית לשינויי נסיבות



דוגמא לתכנית פיתוח באיטרציות

ID	Task Name	Oct 2005	Nov 200	Dec 2005			Jan 2006						
		10/3 10/10 10/17 10/24	10/31 11/7 11/14	11/21	11/28	12/5	12/12	12/19	1/2	1/9	1/16	1/23	
1	Discovery												
2	Iteration 1												
3	Iteration 2												
4	Iteration 3												
5	Iteration 4												
6	Deployment												

SCRUM

- גישה המבוססת על צוותי פיתוח קטנים לפיתוח מערכת באיטרציות כאשר הדרישות משתנות לעיתים תכופות
- הגישה שמה דגש על מחויבות ואחדות
 הצוות, יעילות העבודה ואיכות התוצרים של
 הצוות



SCRUM

- מקורו של המונח "Scrum" הוא במשחק הרוגבי, שם הוא מתאר את הדרך שבה המשחק מתחיל מחדש לאחר שהכדור יצא מהמגרש
 - בפרויקט Scrum תהליך הפיתוח מתחיל שוב מראשיתו מידי חודש. כלומר, פעם בחודש מסופקת תוכנה עובדת ללקוח, והערותיו, כמו גם הדרישות החדשות, מיושמות במחזורי הפיתוח הבאים

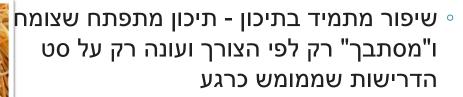
Extreme Programming (XP)

- מתודולוגיית פיתוח תוכנה שנוצרה על ידי קנט בק.
 השיטות המשמשות אותה הן מחמירות מאד, ובעת פרסומה נחשבו כקיצוניות יחסית לשיטות הקיימות בתעשייה לפיתוח תוכנה
- Test Driven Development TDD כתיבת דרישות המערכת כסט של בדיקות הניתנות להרצה, ופיתוח הבדיקות קודם לפיתוח הפונקציונליות.
- ⊙ שיטה זו דורשת הבנה טובה של עקרונות
 סובה של עקרונות בורשת הבנה טובה שיטה זו דורשת הבנה טובה של עקרונות

עקרונות ה- XP

- המעגל הפנימי:
- תכנות בזוגות מתכנת אחד כותב את הקוד,
 והשני מייעץ לו לגבי דרך העבודה, ואחראי על
 איתור תקלות בקוד (תחביריות, לוגיות וכו')
- הזוגות מתחלפים מדי פעם בזוגות אחרים, כך שכל המתכנתים יכירו כמה שיותר קוד
- עיצוב פשוט עמידה בכל הדרישות, ללא קוד
 כפול, הקוד מובן למתכנתים ויש בו המינימום
 הנדרש של מחלקות ושיטות

XP -עקרונות ה

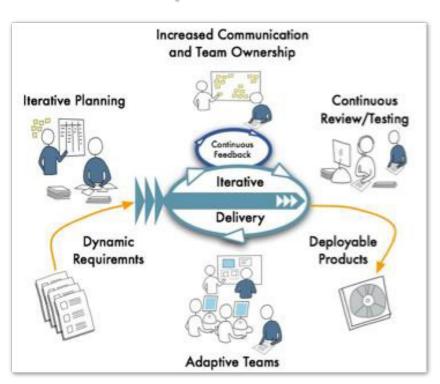


תכנות מונחה בדיקות - כתיבת והרצת בדיקות אוטומטיות כחלק מרכזי בפיתוח, ובתדירות גבוהה

Unit,) הבדיקות מבוצעות בכל הרמות • • httegration, System



?איך זה עובד



בדיקות ב- Agile

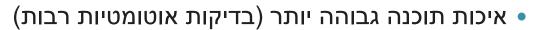
- בודקים ומפתחים יושבים ועובדים ביחד בצוותיםמעורבים
 - תקשורת ישירה עם המפתח והלקוח (ולא דרך מסמכים)
- ∘ STP קצר, ממוקד ב- Sprint הנוכחי, מתפתח דרך דיונים קצרים וממוקדים בנושאים ספציפיים
 - ∘ מכיל רשימה של הערות לצורך בדיקות של דרישות המפותחות ב- Sprint

בדיקות ב- Agile

- דיווח (Status report) דיווח באמצעות תרשים גדול, ממוקם במקום ציבורי, קל להבנה, העונה על השאלה:
 - לגבי איזה חלקים של המערכת אנו יכולים להפסיק ∘ "לגבי איזה חלקים של המערכת אנו יכולים להפסיק ∘ "לדאוג?"
 - הבדיקות יכולות להתחיל עוד לפני שהפיתוח הסתיים
 - הבודק מסייע למפתח בתהליך הפיתוח באמצעותהבדיקות







תוצרים יותר צפויים •

שביעות רצון לקוח גבוהה יותר •

יצרניות (Productivity) הפיתוח גבוהה יותר

• עלות הפרויקט נמוכה יותר (פחות תקלות, פחות תיקונים)





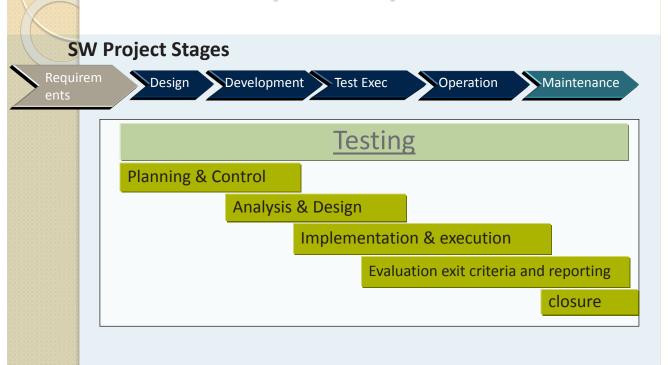
סוגי ארגונים להם מתאים פיתוח לפי Agile

צוותים מוכשרים ויושבים ביחד (פחות מפתחים אך • (מיומנים יותר

- לקוח מחויב, מוכשר ומעורב בתהליך
 - שינויי דרישות דחופים ותכופים •
- התרבות הארגונית תומכת במשא ומתן
 - אמון באנשים •
- מערכות שאינן קריטיות (אינן מסכנות חיים) •



תהליך הבדיקות הבסיסי



מאפייני בדיקות טובים בכל מודל פיתוח

- עבור כל שלב בתהליך הפיתוח ישנה רמת בדיקות
 הבאה לוודא אותו
- עבור כל רמת בדיקות ישנן מטרות ספציפיות לאותה
 רמה
- פיתוח תסריטי הבדיקות בכל רמה מבוסס על תוצרי
 השלב
- כתיבת הבדיקות ותכנונם מתרחשות במקביל לאותו
 שלב בפיתוח התוכנה, על מנת לא לעכב את הפרויקט
 - יש חשיבות רבה לתכנון הזמן בבדיקות •
 - הבודקים צריכים להיות שותפים בתהליך ה-Review
 של המסמכים/התוצרים הבלוונטיים

מאפייני בדיקות טובים בכל מודל פיתוח

- שלבי הבדיקה מאורגנים בהתאם לסוגהפרויקט ודרישותיו
- לדוגמא: בשילוב מוצר מדף (COTS) בשילוב מוצר רוכש (commercial off-the-shelf המוצר יכול לבצע בדיקות אינטגרציה של מוצר המדף עם המערכת המפותחת ברמת בדיקות (System Test)

תרגיל

תרגיל 1 בחוברת תרגילים: בדיקות תוכנה
 ומודלים לפיתוח





- a) All document reviews involve the development team.
- b) Some, but not all, development activities have corresponding test activities.
- c) Each test level has test objectives specific to that level.
- d) Analysis and design of tests begins as soon as development is complete.

2. What is the objective of debugging?

i To localise a defect.

ii To fix a defect.

iii To show value.

iv To increase the range of testing.

- A. i, iii.
- B. ii, iii, iv.
- C. ii, iv.
- D. i, ii.

3. Which of the following are valid objectives for testing?

- i. To find defects.
- ii. To gain confidence in the level of quality.
- iii. To identify the cause of defects.
- iv. To prevent defects.
- A. i,ii, and iii.
- B. ii, iii and iv.
- C. i, ii and iv.
- D. i,iii and iv.

4. Match every stage of the software Development Life cycle with the Testing Life cycle:

- i. Hi-level design a. Unit tests
- ii. Code b. Acceptance tests
- iii. Low-level design c. System tests
- iv. Business requirements d. Integration tests
- a) i-d, ii-a, iii-c, iv-b
- b) i-c, ii-d, iii-a, iv-b
- c) i-b, ii-a, iii-d, iv-c
- d) i-c, ii-a, iii-d, iv-b

5. Which of the following will be the best definition for Testing :-

- a) The goal / purpose of testing is to demonstrate that the program works.
- b) The purpose of testing is to demonstrate that the program is defect free.
- c) The purpose of testing is to demonstrate that the program does what it is supposed to do.
- d) Testing is executing Software for the purpose of finding defects.

6. Which of the following is not a type of incremental testing approach?

- a) Top down
- b) Big-bang
- c) Bottom up
- d) Functional incrementation.

תשובות:

שאלה	Α	В	С	D
1			✓	
2				✓
3			✓	
4				✓
5				✓
6		✓		

