

מבוא להנדסת תוכנה

אוהד ברזילי
אוניברסיטת תל אביב

”אפילו מתכנתים יכולים להיות אנשים
שלמים בעולם האמיתי. XP הוא הלדמנות
לבחון את עצמך, להיות עצמך, להבין
שאוףי היית בסדר כל הזמן וכשאתם הסתובבת
עם האנשים הלא נכונים...”

Kent Beck, eXtreme Programming Explained

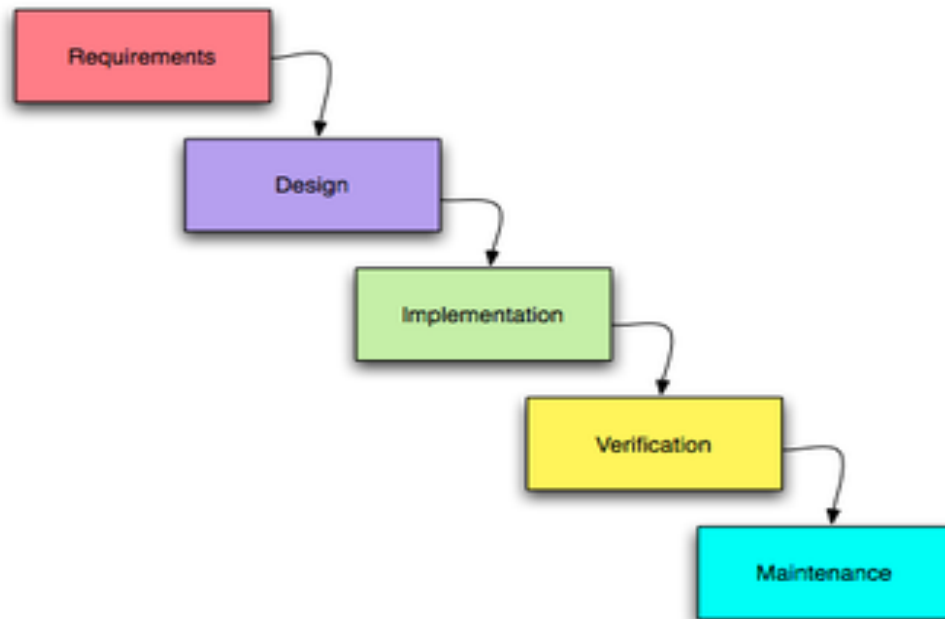
מחזור החיים של תוכנה

- ניתוח דרישות (requirements analysis)
 - תיכון (design)
 - מימוש (Construction, implementation or coding)
 - שילוב (integration)
 - בדיקות וניפוי שגיאות (Testing and debugging aka: verification)
 - בדיקות קבלה
 - ייצור (production)
 - הפצה והתקנה (deployment and installation)
 - תחזוקה ושינויים (maintenance)

 - התייחסות מיוחדת למקרה שמערכת התוכנה היא חלק ממערכת ממוחשבת הכוללת חומרה ותוכנה
-

מודל המפל

□ המודל המסורתי של מחזור חיים נקרא מודל מפל המים (waterfall model, Royce 1970) - כל שלב מתבצע לאחר שקודמו הסתיים (אך ניתן לחזור לשלב קודם לצורך תיקון).



מחירן של טעויות

□ ככל שטעות מתגלה מוקדם יותר, מחיר תיקונה קטן יותר

□ נניח שטעינו בניתוח הדרישות ושכחנו פעולה מסוימת שהתוכנה צריכה לבצע

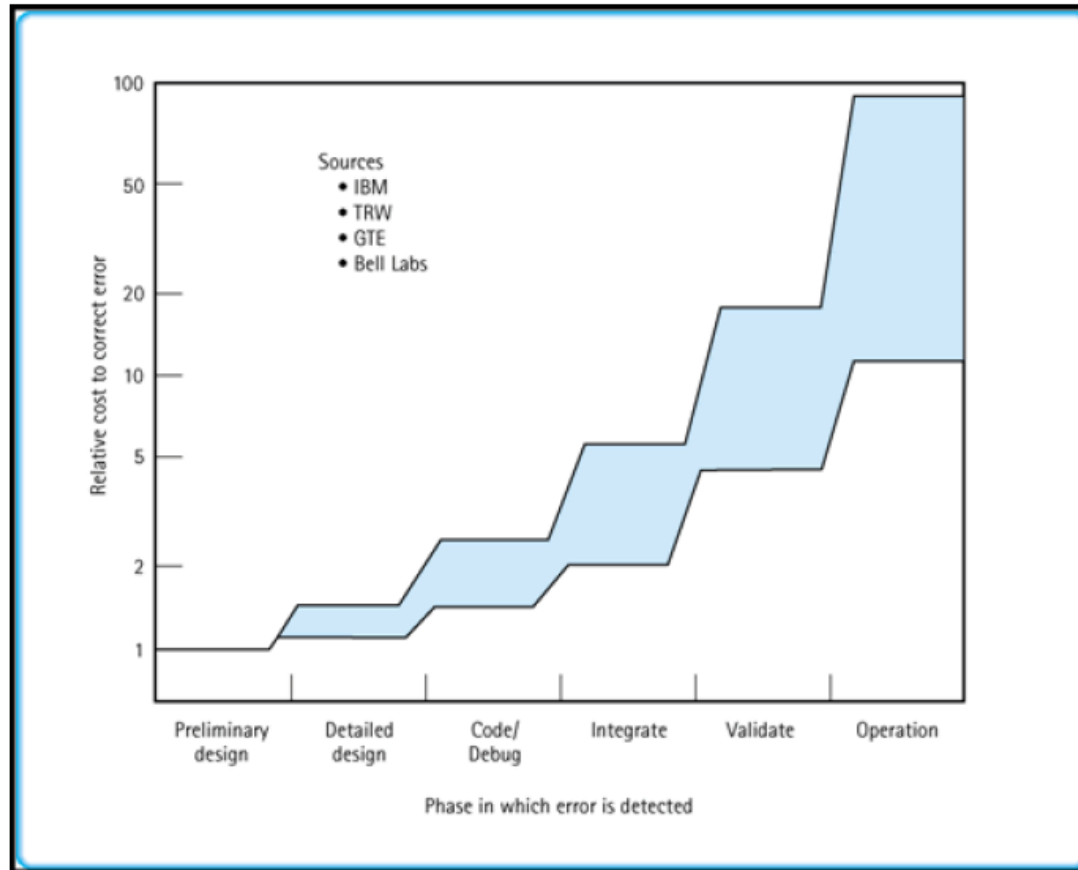
■ אם נגלה את הטעות לפני המעבר לתיכון, המחיר יהיה מינימאלי, אולי עיכוב קטן בלוח הזמנים

■ אם נגלה בזמן התיכון, נצטרך אולי לזרוק חלק מהתיכון שלא יתאים לדרישות המתוקנות

■ אבל אם נגלה את הטעות רק בזמן בדיקות הקבלה, נצטרך אולי לזרוק חלקים גדולים מהתיכון ומהמימוש!

□ עדיף לגלות טעויות מוקדם; לשם כך צריך לתכנן בקפדנות את תהליך הפיתוח הכולל, ולהשתדל להשתמש בשיטות שימזערו טעויות ואת הצורך לחזור אחורה לשלב קודם

מחירן של טעויות





How the customer explained it

לא תכנות. פיתוח תוכנה

□ פיתוח תוכנה אינו כולל רק תכנות, זהו תחום גדול ועשיר בעל הבטים רבים

□ אתגרים בפיתוח תוכנה מודרנית:

- טכנולוגיות משתנות: שפות, סביבות, כלי פיתוח, פלטפורמות
- כמויות אדירות של קוד קיים
- התממשקות לרכיבי צד שלישי
- עבודה עם קוד פתוח
- כתיבת קוד עבור לקוחות ועבור משתמשים
- תוכנה מפותחת ע"י ארגונים
- אתגר הגודל

הגודל כן קובע!

גדול זה קשה

בפיתוח של "תוכנה
אמיתית" מתגלות
בעיות שלא קיימות
בתוכניות מחשב
קטנות

גודל בכמה מישורים:

גודל התוכנה (שורות קוד) ■

זמן ■

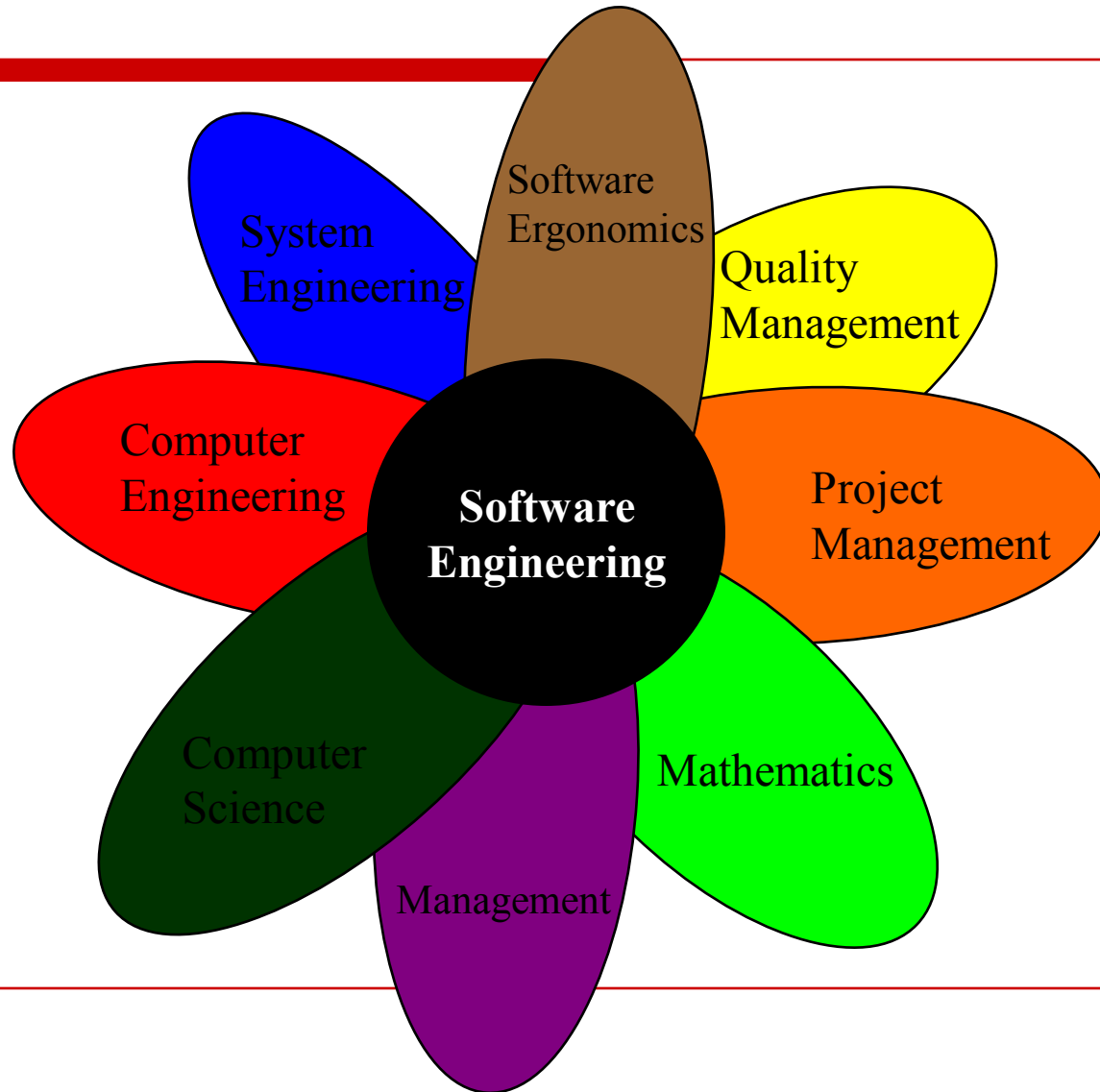
משך פיתוח

חיי מוצר ארוכים

גודל הארגון המפתח ■

גודל אוכלוסיית היעד ■

RELATED DISCIPLINES OF SOFTWARE ENGINEERING (SWEBOK2004)



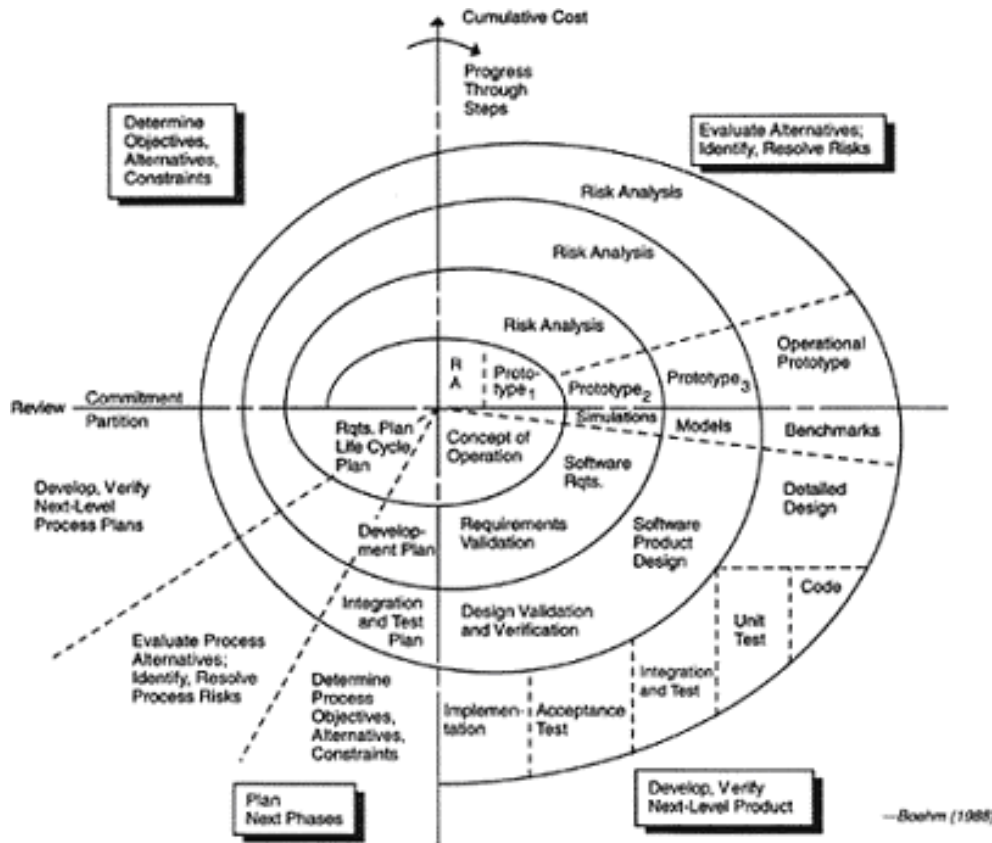


שינוי

Everything in software changes. The requirements change. The design changes. The business changes. The technology changes. The team changes. The team members change.

The problem isn't change, because change is going to happen; the problem, rather, is our inability to cope with change.

מודל ספירלה



מודל הספירלה (spiral)
model (Barry Boehm, 1988)
שהוצע מאוחר יותר
מפתח את המערכת באופן
אבולוציוני.

מתחילים מפיתוח מערכת
מינימלית, ומבצעים את כל
השלבים. לאחר סיום מערכים
את המוצר הנוכחי, מחליטים
מה להוסיף, וחוזרים על כל
השלבים

מפל או ספירלה?

- מודל הספירלה מאפשר לראות מוצר חלקי ולהעריך אותו
- אבל מפל המים משקף את הרצוי: רצוי לא לטעות.

- קיימים גם מודלים אחרים לתהליך הפיתוח:
 - בשנים האחרונות עולה הפופולריות של משפחת המודלים הקלילה (agile) וכן של ייצור רזה (lean manufacturing)
 - נציג בולט של המשפחה הזו הוא **eXtreme Programming** (תכנות קיצוני, XP)

תכנות קיצוני (XP)

מעצבי השיטה (Kent Beck, Ward Cunningham, Ron Jeffries, 1996) ניסחו 5 ערכים:

■ משוב (feedback)

■ פשטות (Simplicity)

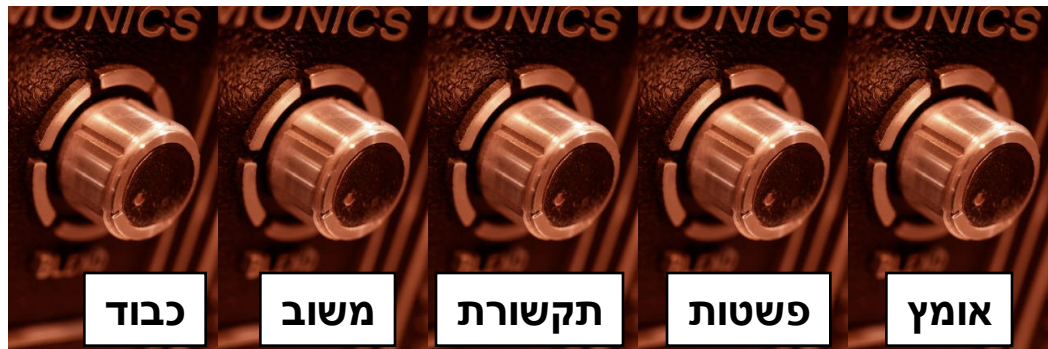
■ תקשורת (Communication)

■ אומץ (Courage)

■ כבוד (Respect)

מכיוון שהערכים הוכחו כטובים, נלקח כל אחד מהם לקיצוניות

ערכים, עקרונות, מיומנויות



- ערכים אלו כלליים ויש לגזור מהם **מיומנויות**, דברים פשוטים שנוכל ליישם ברמה הטכנית
- חשוב להבין את **העקרונות** שלפיהם נגזרו המיומנויות כדי שנוכל להתאים את השיטה לצרכנו (להוסיף או להסיר מיומנויות)

העקרונות

- הזדמנות (opportunity)
- יתירות (redundancy)
- כשלון (failure)
- איכות (quality)
- צעדי תינוק (baby steps)
- קבלת אחריות (accepted responsibility)
- אנושיות (humanity)
- כלכלה (economics)
- רווח הדדי (mutual benefit)
- דמיון עצמי (Self Similarity)
- שיפור (Improvement)
- שוני, מגוון - (diversity)
- הרהור (reflection)
- זרימה (flow)

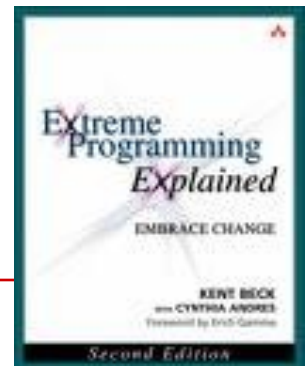
אבולוציה של מיומנויות

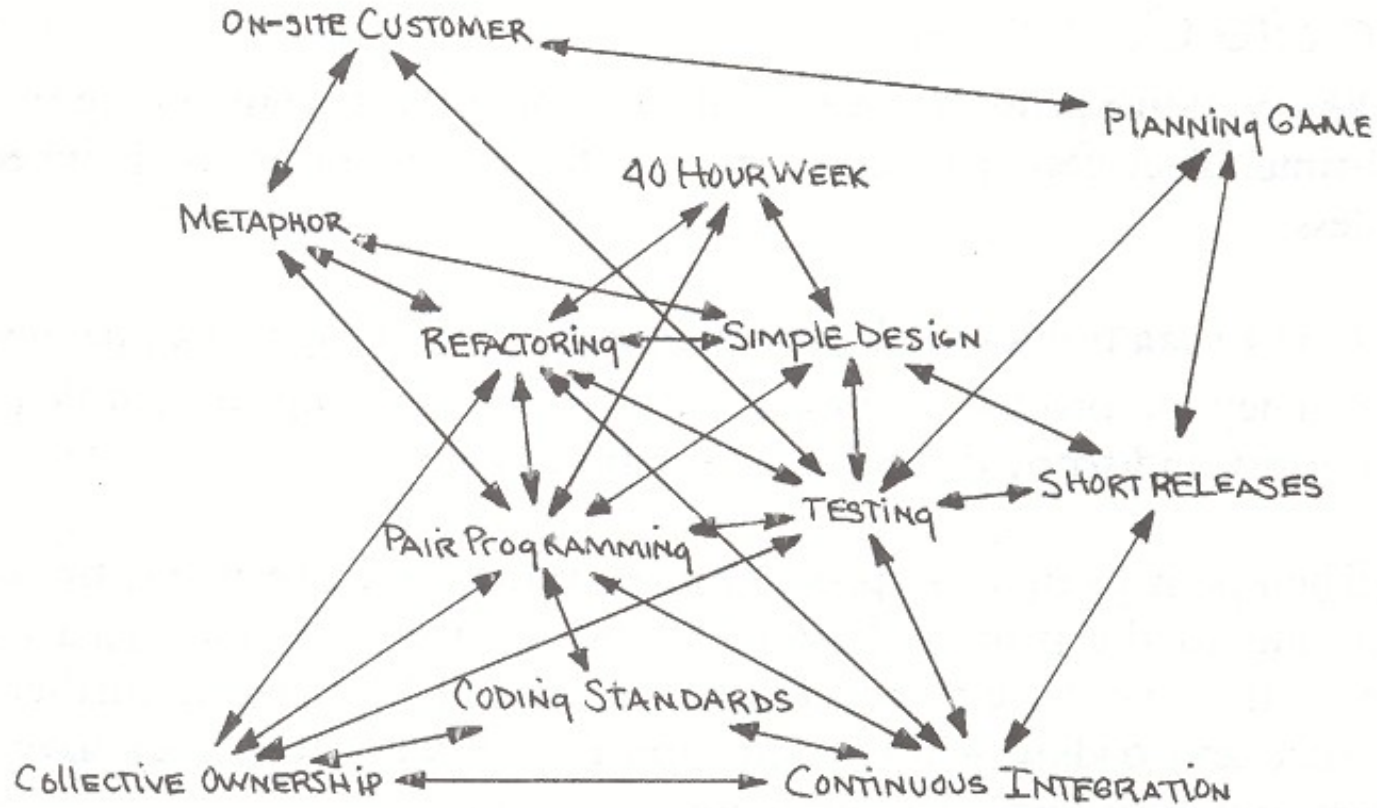
□ ניתן ללמוד על ההבשלה של גישת XP ע"י בחינת השינויים שעברו המיומנויות בין המהדורות של הספר [במצגת נפרדת]

- ***Extreme Programming Explained***

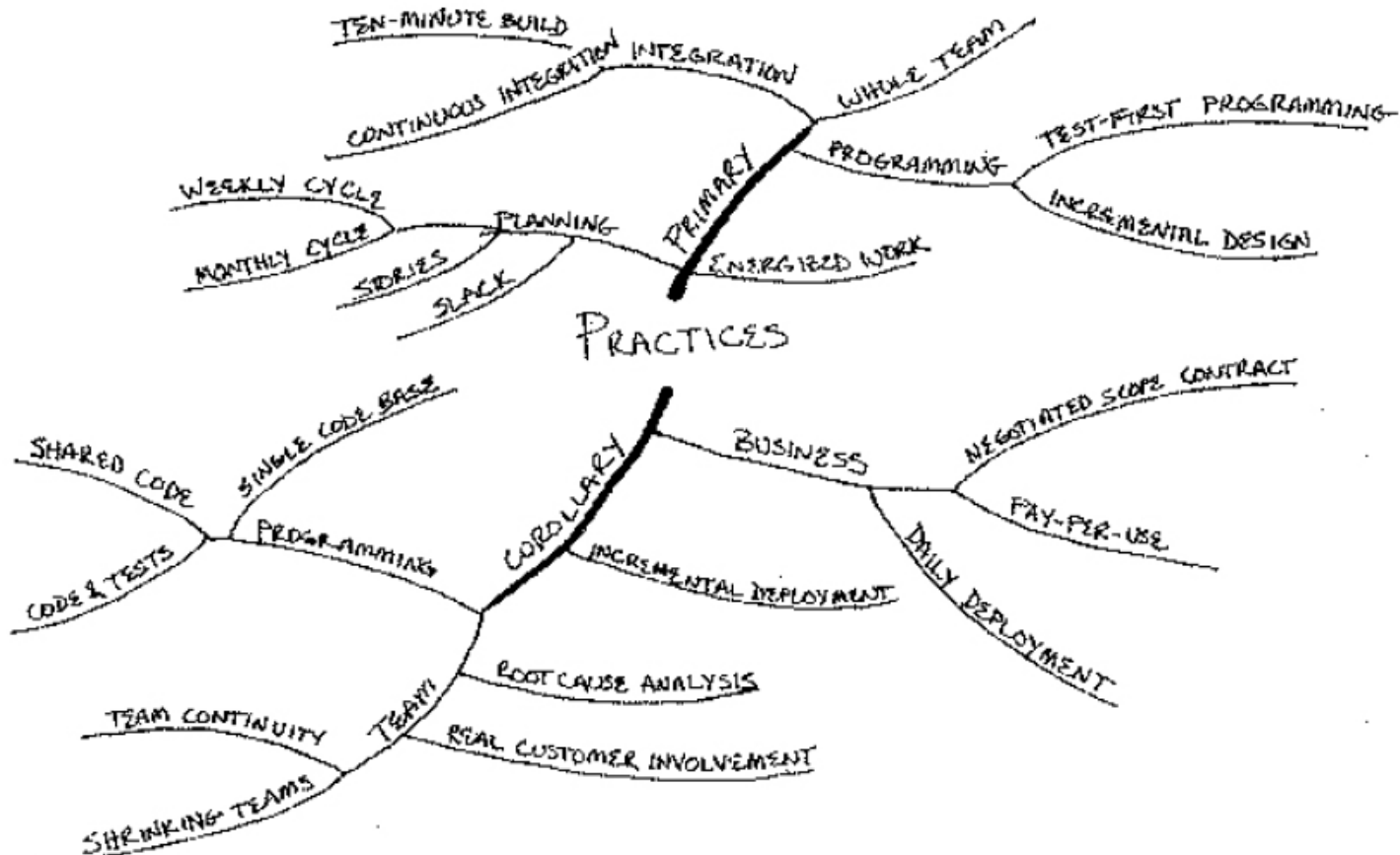
- **First Edition, Kent Beck, 2000**

- **Second Edition, Kent Beck with Cynthia Andres 2005**





Source: Beck, K. (2000). *eXtreme Programming explained*, Addison Wesley.



Source: **eXtreme Programming explained**
 Kent Beck with Cynthia Andres, Second Edition, 2005