**סיכום פרק 9 – התפתחות תוכנה**

**26 .מדוע אבולוציית מערכת חשובה?**

שינויי תוכנה בלתי ניתנים לעצירה:

דרישות חדשות צצות תוך כדי שימוש

סביבת העבודה משתנה

טעויות חייבות להפתר

חומרה חדשה מתבספת למערכת

שיפור אמינות המערכת

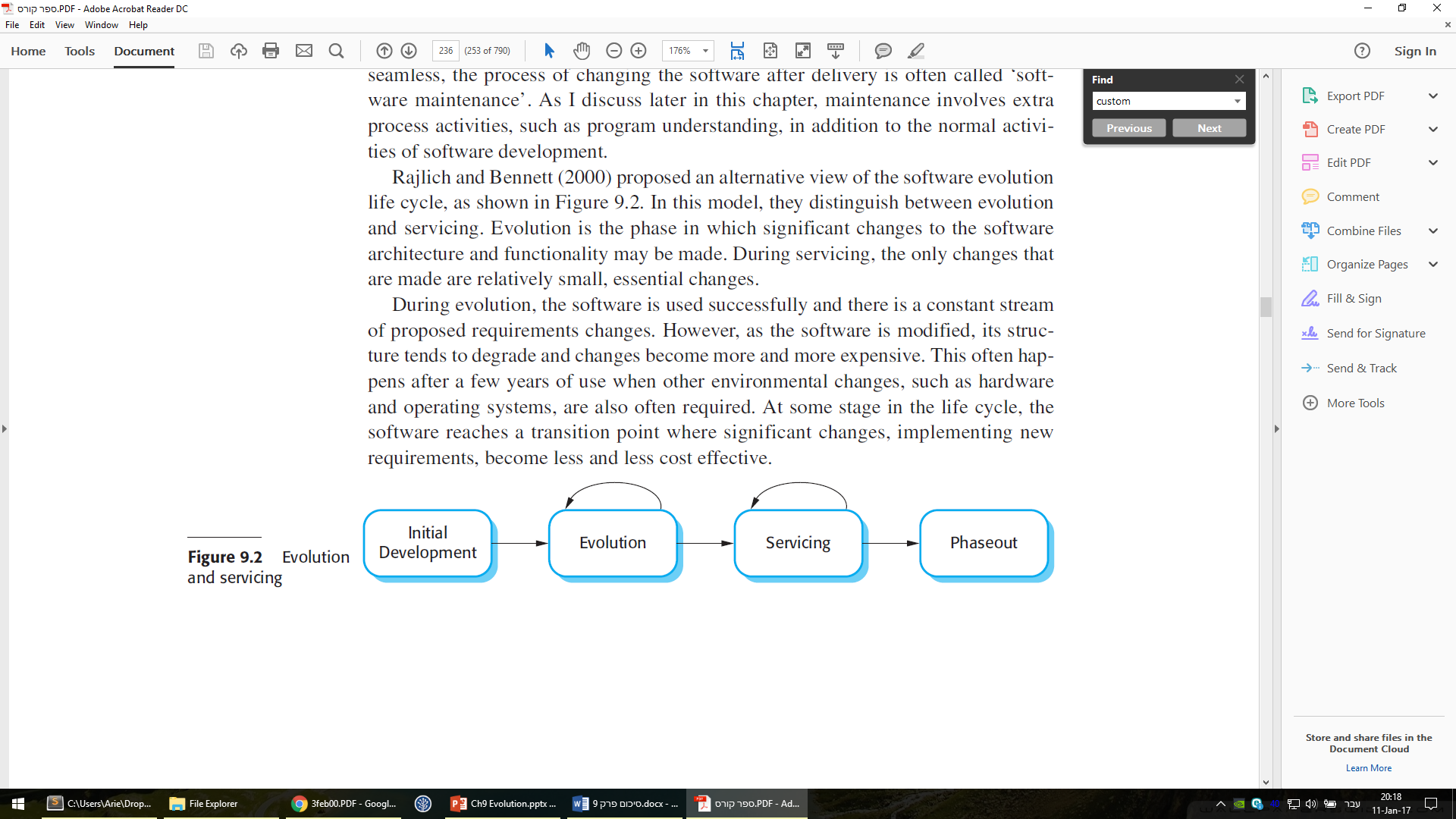
חברות משקיעות הון במערכות התוכנה שלהן, וכדי לשמור על ערכן של מערכות אלו קיים צורך קריטי בעדכון ושינוי שלהן. אחוז עיקרי מההשקעה במערכות אלו הינו לחידוש ושידרוג, ולא על פיתוח חדש.

**27 .מהו התהליך הספיראלי של פיתוח ואבולוציה, אילו שלבים כולל התייחס למוצר גנרי ומוצר מותאם אישית**

**ללקוח? מה מחדשים Bennett and Rajlich בתהליך?**

9.1 SpiralEvolution.eps

* **התהליך הספיראלי מתאר את "המצבי" מערכות תוכנה, מהגדרת דרישות, מימוש, ווידוא ותיקוף ועבודה בפועל, כאשר שלב הדרישות מתייחס למוצר מותאם אישית, מול של האימות והתיקוף המתייחס למוצר גנרי**
* **עפ Bennett and Rajlich יש להפריד את "התפתחות" ל"תחזוקה":**



* + **אבולוצייה:** שלב בו התוכנה בשימוש ומתעדכנת תוך כדי עקב דרישות חדשות
  + **תחזוקה:** שלב בו התוכנה בשימוש אך השינויים מתבצעים אך ורק כדי שתתפקד (למשל תיקון באגים)
  + **יציאה משירות:**  שלב בו התוכנה עדיין בשימוש עקב חוסר ברירה אך לא מתבצעים שינויים כלל

**28 .תאר את תהליך ההתפתחות, מהם השלבים?**

התפתחות התוכנה תלוייה ב:

* סוג התוכנה
* צורת הפיתוח
* רמת המקצועיות של המפתחים

הצעות לשינויים נובעות מהתפתחות התוכנית; שינויים אלה צריכים להיות חלק אינטגרטיבי במערכת, כדי שיהיה אפשר לשאר את עלותם ותועלתם.

9.4 Evolution Process.eps

**29. במה שונה התהליך בסעיף 28 מתהליך פיתוח תוכנה כללי?**

בפיתוח תוכנה כללי, יש יחס לדרישות, פיתוח, ווידוא ותיקוף דרישות ויישום, אך אין דגש על שינויים והתפתחות התוכנה בעתיד. התהליך בסעיף 28 מפרט את מהלך חיי התוכנית.

**30. מה מייחד שינוי דחוף, כיצד מתבצעת ההטמעה ומה ההשלכות?**

**9.6 EmergencyRepair.eps**

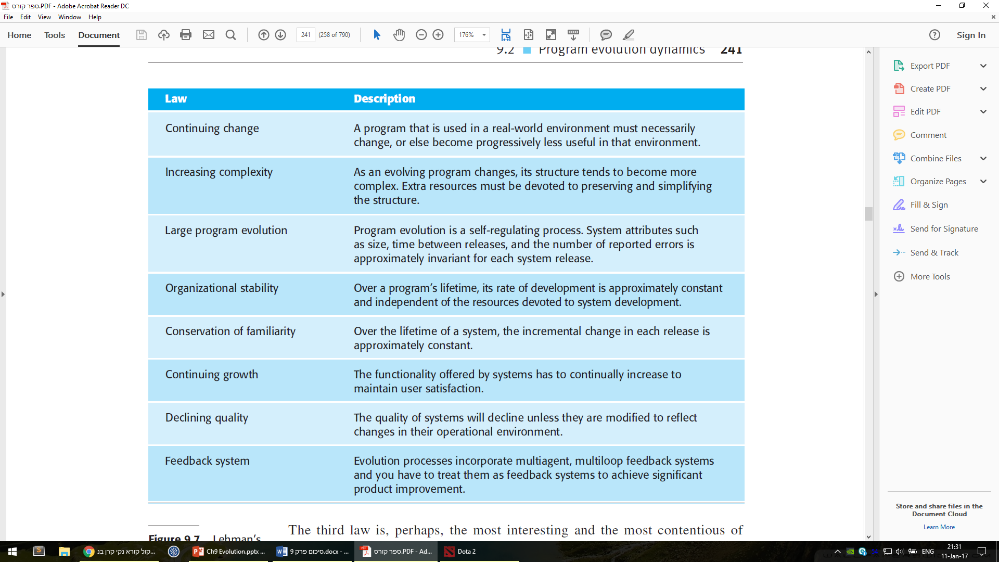
שינוי דחוף אינו דורש בדיקות "כדאיות" ובדיקת דרישות, המטרה ברורה: לתקן בדחיפות

**31 .מדוע למערכות מותאמות custom העלויות של שלב האבולוציה עולות פעמים רבות על עלות פיתוח**

**התוכנה?**

פיתוח מעררכות מותאמות אישית יעלו לפעמים פעמים רבות יותר עקב העובדה כי הלקוח הוא בעלי התוכנה, ולעיתים ירצה שבית תוכנה אחר יעבוד על התוכנה שלו, או שחברות מתמזגות או פושטות רגל, ולכן בית התוכנה החדש יצתרך ללמוד את התוכנה של הלקוח, דבר אשר יגלגל עלויות על הלקוח עצמו

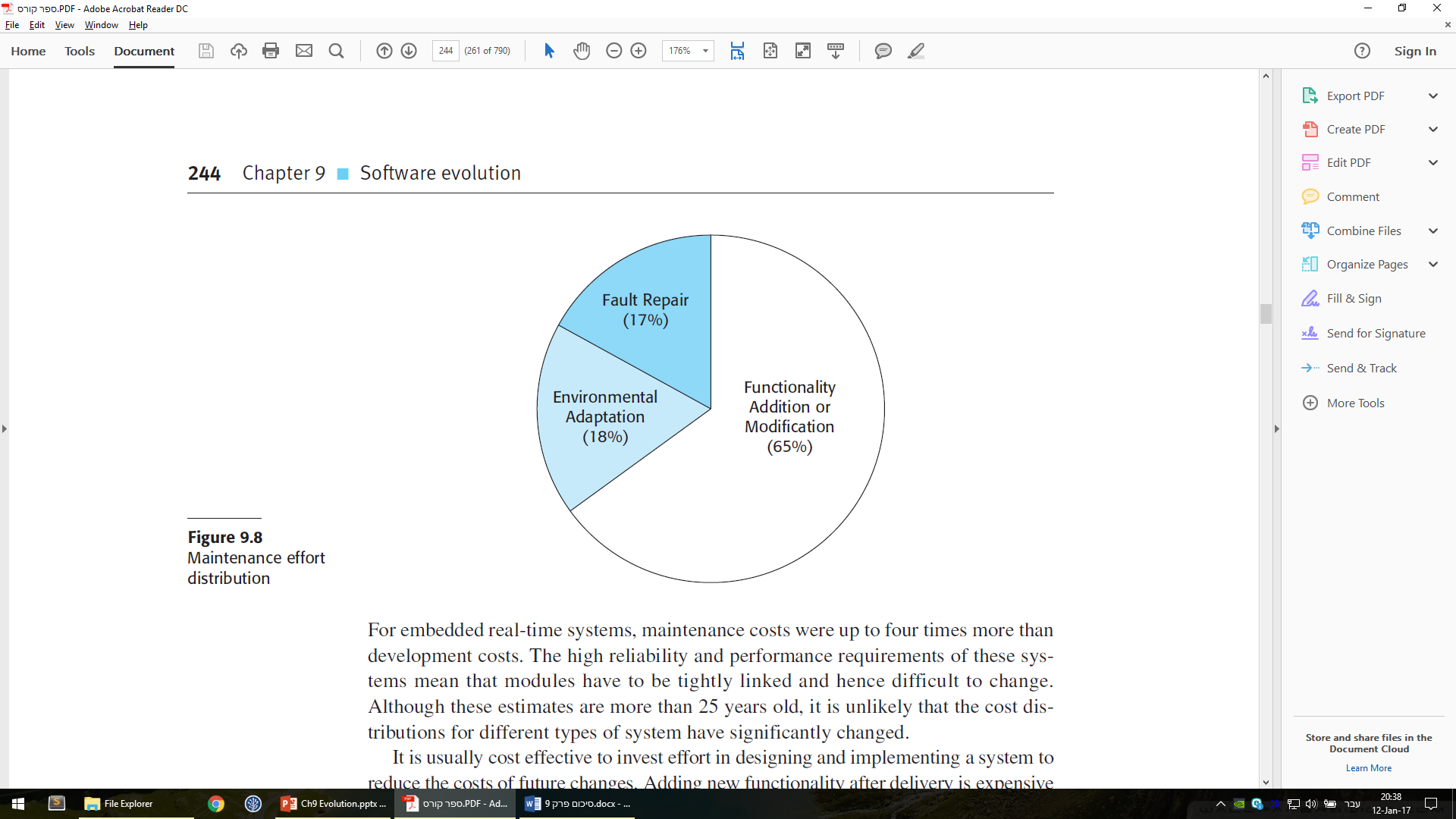
**32 .הסבר את התהליך של האבולוציה – מה ההתחלה ומהן ההשלכות.**

תהליך האבולוצייה בפיתוח גמיש מתבסס על פיתוח אינקרמנטאלי, כאשר המעבר לשלב האבולוצייה לא מורגש, מאחר והוא המשך פיתוח ושיחרור גרסאות כמו שהתצע עד כה.

**33 .מהם חוקי להמן laws s’Lehman ?באילו נסיבות לא יתקיימו?**

אלו הם חוקים הנוגעים לשינויים במערכות. חוקים אלו תקפים לרוב בחברות גדולות.

לאומת זאת חוקים אלו לא בהכרח חלים על חברות קטנות, בהן שינויים לא נובעים עקב דרישה לשינוי תדמית/צורת עבודה של החברה



**34 .מהם שלושת סוגי תחזוקת התוכנה? מדוע קשה לעיתים להפריד ביניהם?**

# שגיאות קוד

# הסתגלות/התאמה לסביבה חדשה

# הוספת פונקציונאליות/דרישות

בפועל קשה להפריד בין שלושת הסוגים, למשל במצב של שידרוג סביבת מערכת, תיבסף שימושיות שלא הייתה קיימת, ועקב זה גם יכולות להופיע שגיאות קוד.

**35 .מהו תהליך refactoring?**

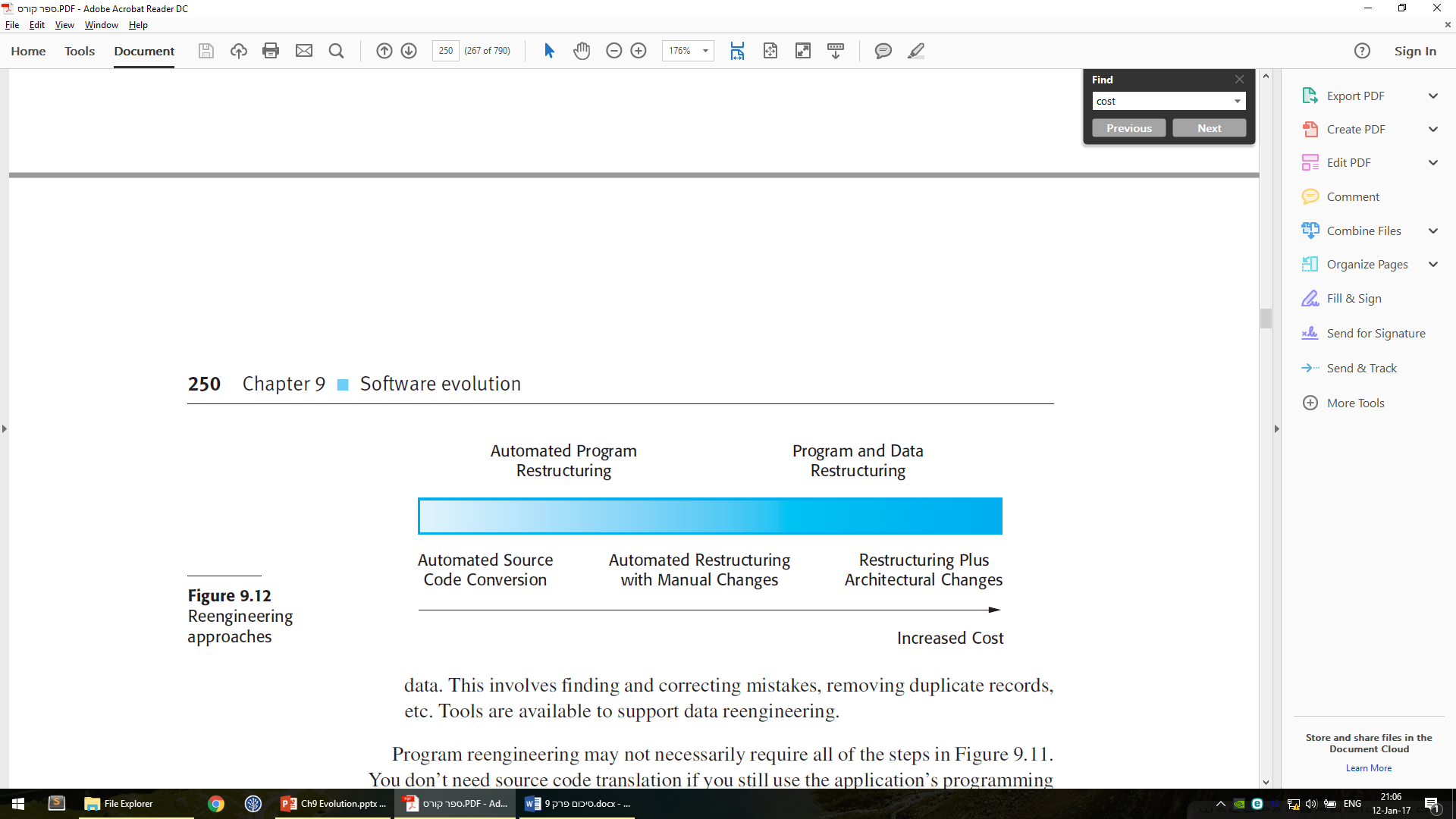
* **Refactoring is the process of making improvements to a program to slow down degradation through change.**
* **Refactoring is a continuous process of improvement throughout the development and evolution process. It is intended to avoid the structure and code degradation that increases the costs and difficulties of maintaining a system.**

**36 .בהנחה כי הנך מנהל את התוכנה של הפרויקט שבקורס. הצע שיטות לאיתור הגורמים שישפיעו על אבולוציית התוכנה.**

תיכנון מראש – יאפשר זיהוי מצבים בהם התכונה תתפתח/תשתנה.

בדיקות יחידה – יאפשרו זיהוי מהיר של שגיאות.

**37 .מהם הגורמים המרכזיים המשפיעים על עלויות הנדסה לאחור reengineering?**



**38 .האם קיימים מצבים בהם מחליטים להחליף את המערכת למרות שמדדי האיכות מראים איכות וערך עסקי גבוהים?**

**מדוע?**

The business value of a legacy system and the quality of the application software and its environment should be assessed to determine whether the system should be replaced,transformed, or maintained.