**סיכום 2**

**קבוצה מס"44 :** אלגואברה פרס : 319146684

עואודה באסל : 316573005

1. **מה חשיבות עיצוב הארכיטקטורה בתוכנה?**

הוא שלב מוקדם של תהליך עיצוב מערכת ומייצג את הקשר בין תהליכי אפיון ועיצוב , והפלט של תהליך עיצוב זה הוא תיאור של ארכיטקטורת התוכנה.

אדריכלות בגדולה היא עוסקת בארכיטקטורה של מערכות ארגוניות מורכבות הכוללות מערכות אחרות, תוכניות, ורכיבי תכנית. מערכות הארגוניות אלה מופצות על מחשבים שונים, אשר עשוי להיות בבעלות ומנוהלים על ידי חברות שונות.

1. **מהם 3 היתרונות העיקריים בתיכנון, עיצוב ותיעוד הארכיטקטורה? הסבר גם את החיסרון על פי Bass.**

\*תקשורת בעלי העניין

אדריכלות יכולה לשמש כמוקד של דיון על ידי בעלי עניין במערכת.

\*ניתוח מערכת

משמעות הדבר היא שהאם הניתוח הראשי למערכת יכול לעמוד בדרישות שאינן פונקציונליות שלה הוא אפשרי.

שימוש חוזר

הארכיטקטורה עשויה להיות לשימוש חוזר במגוון של מערכות

ארכיטקטורות המוצר אונליין יכולות להיות מפותחות.

1. **הסבר מהו תרשים block diagram, כיצד תורם לתהליך העיצוב. כמו כן ציין את המגבלות של תרשים זה.**

תרשים זה מראה גופים ומערכות יחסים שהם השיטות הנפוצות ביותר בשימוש לתיעוד ארכיטקטורות תוכנה. אבל הם צריכים ביקורת כי חסר להם סמנטיקה .

לא מראה סוגים של מערכות יחסים בין גופים ולא את המאפיינים הגלויים של גופים באדריכלות , והוא תלוי בשימוש במודלים אדריכליים.הדרישות למודל סמנטיקת תלויות באופן שימוש בדגמים.

1. **תאר את ההחלטות שאותן יש לבצע במהלך עיצוב הארכיטקטורה- תן דוגמה ממערכת )שאתה מכיר או מהמערכת של הפרויקט( כיצד בא לידי ביטוי.**

עיצוב אדריכלי הוא תהליך יצירתי ולכן התהליך משתנה בהתאם לסוג מערכת מפותח.

עם זאת, מספר ההחלטות משותפות להקיף את כל תהליכי התכנון וההחלטות האלה משפיעות על המאפיינים שאינם הפונקציונליים של המערכת. והשאלות הן :  
האם יש ארכיטקטורת יישום כללית שניתן להשתמש בו?  
איך המערכת תופץ?  
מה אדריכלי סגנונות מתאימים?  
מה גישה ישמש למבנה המערכת?  
איך המערכת להיות מפורקת למודולים?  
מה שליטת אסטרטגיה יש להשתמש?  
איך העיצוב האדריכלי להעריך?  
איך הארכיטקטורה צריכה להיות מתועדת?  
דוגמה : מערכת לתמיכת החלטה קלינית לאבחון רפואי .

1. **הסבר מהי תבנית ארכיטקטורה, הסבר את סוגי התבניות הבאים שרת –לקוח, MVC, ארכיטקטורת שכבות, ארכיטקטורת מאגר, Pipe&Filter תן דוגמה לכל סוג.**

תבניות הם אמצעי לייצוג, שיתוף ושימוש חוזר בידע.

תבנית אדריכלית היא תיאור מסוגנן של תרגול עיצוב טוב, שכבר ניסה ונבדק בסביבות שונות.

תבניות צריכים לכלול מידע על כאשר הם וכאשר הם לא שימושיים.

דפוסיםתבניות יכולים להיות מיוצגים באמצעות תיאור טבלאי וגרפי.

**The Model-View-Controller (MVC)\*** : היא סוג תבנית , מצגת מפרידה ואינטראקציה מנתוני המערכת , משמש כאשר יש מספר דרכים להציג ולעבוד עם הנתונים. דוגמה : מערכת מבוססת אינטרנט .

**\*** ארכיטקטורת שכבות :

מארגן את המערכת לשכבות עם פונקציונליות קשורה לכל שכבה , שימוש בעת בניית מתקנים חדשים על גבי מערכות קיימות; כאשר הפיתוח היא התפשטה על פני כמה קבוצות עם כל אחריות צוות לשכבה פונקציונליות; כאשר יש דרישה לאבטחה רבת-שכבתית . דוגמה : מודל שכבתי של מערכת לשיתוף מסמכי זכויות יוצרים שהתקיימו בספריות שונות .

\*ארכיטקטורת מאגר : כל הנתונים במערכת מנוהלים במאגר מרכזי שנגיש לכל רכיבי המערכת. רכיבים לא לתקשר ישירות, רק דרך המאגר.

משתמשים בתבנית זו כאשר יש מערכת שבה כמויות גדולות של מידע נוצרות שיש לאחסן במשך זמן רב . דוגמה: כל כלי תוכנה מייצר מידע אשר לאחר מכן זמין לשימוש על ידי כלים אחרים.

\*ארכיטקטורת שרת-לקוח : בארכיטקטורת שרת-לקוח, פונקציונלי של המערכת מאורגנת בשירותים, עם כל שירות שסופק משרת נפרד. לקוחות הם משתמשים בשירותים אלה ושרתי גישה ללעשות בהם שימוש. משמש כאשר נתונים במסד נתונים משותפים יש לגשת ממגוון מקומות . דוגמה : ספריית סרטים ו- DVD / וידאו המאורגן כמערכת שרת-לקוח. **pipe and filter** : העיבוד של נתונים במערכת מאורגן כך שכל רכיב עיבוד (מסנן) הוא בדיד ומבצע סוג אחד של שינוי הנתונים.נתונים זורמים (כמו בצינור) ממרכיב אחד למשנהו לעיבוד. דוגמה: עיבוד חשבוניות .

1. **מדוע מודל MVC הוא המתאים ביותר למידול מערכות web?**

כי משתמשים ב MVC כאשר יש מספר דרכים להציג ולעבוד עם הנתונים. משתמשים גם כאשר הדרישות העתידיות לאינטראקציה ולהצגה של נתונים אינן ידועות. ועוד מאפשר את הנתונים כדי לשנות באופן עצמאי מייצוגה, ולהיפך. תומך מצגת של אותם נתונים בדרכים שונות עם שינויים שנעשו בייצוג אחד המוצגים בכולם. ובדיוק זה מה ש web דושר בכדי הצגת הנתונים והיכולת לשינוי חוזר .

1. **מהו ?ADL**  
   ADL היא אחד השפות היאור , שפת תיאור (ADLs) מתארת ארכיטקטורות מערכת,האלמנטים הבסיסיים של ADLs הם רכיבים ומחברים, והם כוללים כללים והנחיות לארכיטקטורות בנויות היטב. עם זאת, בגלל מיוחד שלהם מומחי הטבע, תחום והיישום מתקשים להבין ולהשתמש ADLs.זה עושה את זה קשה להעריך את התועלת שלהם להנדסת תוכנה מעשית.ADLS המיועד לתחום מסוים (לדוגמא, מערכות רכב) **.**
2. **מהי ארכיטקטורת יישום Application Architecture?**

מערכות יישום נועדו לענות על צורך ארגוני, כיש הרבה מן משותף עסקים, מערכות יישומם גם נוטות להיות ארכיטקטורה משותפת המשקפת את דרישות היישום.  
ארכיטקטורת יישום גנריות היא ארכיטקטורה לסוג של מערכת תוכנה שיכולה להיות מוגדר ומותאם ליצירת מערכת שעונה על דרישות ספציפיות.

1. **תאר והדגם את שני סוגי המערכות של Application Architecture- מערכת עיבוד טרנזקציות .)language processing systems( שפה עיבוד ומערכות ,)transaction processing systems(**

\*מערכת עיבוד טרנזקציות : כמו מערכות מסחר אלקטרוני ומערכות הזמנה.  
משתמשים לטפל בבקשות לקבלת מידע ממאגר מידע או בקשות לעדכון מסד הנתונים. מנקודת מבט של משתמש עסקה היא:  
כל רצף עקבי של פעילות שמקיימת את המטרה ,לדוגמא - למצוא את הזמנים של טיסות מלונדון לפריז.  
משתמשי בקשות אסינכרוני לשירות אשר לאחר מכן מעובדות על ידי מנהל עסקה.  
\*מערכות עיבוד שפה : קבל שפה טבעית או מלאכותית כקלט וליצור כמה ייצוג אחר של אותה שפה. עשוי לכלול מתורגמן לפעול על ההוראות בשפה שנמצא בתהליך עיבוד.  
משמש במצבים שבם הדרך הקלה ביותר כדי לפתור בעיה לתאר אלגוריתם או לתאר את נתוני המערכת .

1. **הסבר את הקשר בין ארכיטקטורת המערכת ודרישות לא פונקציונאליות**

דרישות לא פונקציונאליות עוסקות בניתוח מערכת  
משמעות הדבר היא שהניתוח של האם המערכת יכולה לעמוד בדרישות שאינן הפונקציונליות שלה הוא אפשרי.  
הדרישות לסמנטיקת מודל תלויות באופן שימוש בדגמים.  
כיש הרבה מן משותף עסקים, מערכות יישומם גם נוטות להיות ארכיטקטורה משותפת המשקפת את דרישות היישום.  
קשר הדוק בין דרישות ותוכנה שאינן פונקציונליות לארכיטקטורה, סגנון מסוים אדריכלי ומבנה שאתה בוחר ל מערכת צריכה להיות תלויה בדרישות המערכת שאינן פונקציונליות כמו :  
האבטחה אם היא דרישה קריטית, מבנה שכבתי לארכיטקטורה יש להשתמש, עם הנכסים הקריטיים ביותר המוגנים בשכבות הפנימיות ביותר, עם רמה גבוהה של אימות אבטחה יחול על שכבות אלה.

1. **מהם ההיבטים הנדרשים לתיאור ועיצוב מערכות תוכנה )ע"פ Hofmeister ו Krutchen(?**

ע''פ Hofmeister ההיבט הוא היבט אדריכלי של מערכת והוא שימושי לתקשורת עם בעלי עניין מערכת ותכנון פרויקט משום שהוא לא עמוס בפרטים. הנוף של ארכיטקטורת תוכנה הידועה של Krutchen, יש לו ארבעה תצוגות אדריכליות, הקשורים באמצעות מקרי שימוש או תרחישים , והנופים הם: 1. מבט הגיוני 2. מבט תהליך,3 . מבט פיתוח 4. מבט פיזי .

1. **. הסבר את החשיבות של תיאור ארכיטקטורת המערכת לפני סיום תהליך פירוט הדרישות ) detailed .( requirements specification**

החישביות של תיאור ארכיטורות המערכת לפני סיום התהליך פירוט דרישות נובע מכך שפירוק דרישות ברמה הגבוהה למפרט מפורט יותר, עוזר למהנדסים לקבל החלטות לגבי רכיבים שניתן לעשות שימוש חוזר, ומייצגים קו מוצרים, ולא במערכת אחת.

1. **צייר תרשימי ארכיטקטורה מהיבט כללי והיבט תהליכי עבור מערכת רובוטית ביתית לניקוי רצפות.**

Clean

floors

Performance

Drop the dust

Electricity

battery

robotic system for home

1. **מהן מטרות תהליכי הבדיקות?**

בדיקות הן כדי לבדוק אם התוכנית עושה מה שהיא אמורה לעשות בדיוק לפני שהיא מפורסמת או יוצאת לשימוש. והמטרות לבדיקה הן:

1. להציג למתכנת או למשתמש שהתוכנית עונה על דרישותיה(התוכנה אמורה לענות על הדרישות ולהיות תקינה).
2. לגלות את המצבים הלא רצויים או מתאימים למפרט של התוכנה(התוכנה אמורה להיות מתאימה לבקשת המשתמש).
3. **הסבר את המשפט של 1972,Dijkstra השתמש במונחי תיקוף ואימות: בדיקות יכולות להראות את הימצאותן של הטעויות, אך לא את העדרן?**

בדיקה יכולה רק להראות הנוכחות של טעויות, לא בהיעדרם

בדיקה היא חלק מתהליך רחב יותר של אימות תוכנה ואימות (V & V).

מַתַן תוֹקֵף ואימות הם לא אותו דבר, למרות שהם מבולבלים לעתים קרובות.

1. **מהם ההבדלים בין חקירת תוכנה לבדיקות תוכנה?**

חקירת תוכנה שונה בגלל שהיא מבוססת על :

הגירסות מושלמות של מערכת יכולות להיבדק ללא עלויות נוספות.

כמו גם מחפש פגמי תכנית, בדיקה יכולה גם לשקול תכונות איכות רחבות של תכנית, כגון עמידה בתקנים,ניידות, תחזוקה

אבל לבדיקות תוכנה מבוססת על דברים אחרים והן :

בדיקות שחרור, שם צוות בדיקה נפרד בדיקות גרסה מלאה שלהמערכת לפני שהוא משוחרר משתמשים.

בדיקות משתמש, שבו משתמשים או משתמשים פוטנציאליים של מערכת לבדוק את המערכת שלהם

סביבה עצמו.

1. **.מה ההבדל בין בדיקות שיחרור לבדיקות משתמש?**

**בדיקת שחרור** : המטרה העיקרית של תהליך בדיקת שחרור היא לשכנע את הספק של המערכת שזה מספיק טוב , היא בדרך כלל תהליך לבדיקת קבוצה שחורה שבו בדיקות נגזרות רק ממפרט המערכת . בדיקת המשתמש: המשתמש בודק את התוכנית לפי מה שהוא מבקש בסביבה שלו.

1. **מהן הבדיקות הקשורות בבדיקות הפיתוח?**

שלבי הבדיקה :

1. בדיקת התפתחות: בדיקה תוך כדי התפתחות התוכנית למצוא בעיות ושגיאות.
2. בדיקת שחרור: צוות בודקים, בודקים את כל התוכנית לפני שמפרסמים אותה למשתמש.
3. בדיקת המשתמש: המשתמש בודק את התוכנית לפי מה שהוא מבקש בסביבה שלו.

אסטרטגיות הבדיקה: לחלק את הבדיקה לקבוצות של קלט זהה שיש להם מאפיינים משותפים ולבדוק אותם באותה דרך. בדיקה המתבוססת על קו-מנחה; משתמשים בזה לבחור מקרים של בדיקה, הנחיות האלה משקפות ניסיון קודם למקרים של שגיאות שהמתכנתים לעתים קרובות עושים אותם תוך כדי פיתוח רכיבים.

* הבדיקה של תוכנית יכולה רק להציג הנוכחות של השגיאות היא לא יכולה להפגין שלא נותר שגיאות. פיתוח הבדיקות הוא האחראי על בדיקת התוכנית לפני שמתפרסמת למשתמש.

בדיקת מערכת ורכיב

1. במהלך בדיקת מערכת ,רכיבים שניתן לעשות בהם שימוש חוזר שפותחו בנפרד , מערכות מוכנים לשימוש באופן חופשי אפשר לשלב אותם על רכיבים שפותחו חדש , ואחר מכן בודקת המערכת השלמה
2. רכיבים שפותחו על ידי חברי צוות שונים או תת-קבוצות יכולים להיות משולבים בזמן זה , בדיקת מערכת קולקטיבית ולא תהליך בודד או אישי .

בדיקת שחרור : המטרה העיקרית של תהליך בדיקת שחרור היא לשכנע את הספק של המערכת שזה מספיק טוב , היא בדרך כלל תהליך לבדיקת קבוצה שחורה שבו בדיקות נגזרות רק ממפרט המערכת ...

1. **האם יש קשר בין תהליכי הבדיקות השונים? הסבר ותן דוגמאות.?**

כן יש קשר

1. **.מהן שתי הגישות הרווחות לגבי טעויות אנוש?**

מתרחשות בשלב יישום המערכת שלב אשר משפיע על כל מהימניות המערכת, יש שני גישות לטעויות אלו:   
-גישת המשתמש, טעויות אשר מתרחשות הן באשמתו של המשתמש והאחריות נטולה עליו גישה זו מתייחסת למשתמשים ומייחסת להם המשקל העיקרי בטעות ולכן הפתרון גם תלוי בהם ולכן לפי גישה זו כדי למנוע טעויות צריך לקבוע ענישות, להגביר המודעות והמקצועיות דרך הגברת הדרכות.  
-גישת המערכת, טוענת שאנשים תמיד חשופים לטעויות, ולכן המערכת מעוצבת כך שתוכל לגלות הטעויות לפני שיתרחשו ויגרמו לקריסת המערכת, לכן לפי גישה זו כשמתרחשת טעות השקעת המאמצים היא לא במשתמש אלא במערכת. בזיהוי למה המערכת לא גילתה הבעיה קודם.

1. **כיצד ניתן לבחור את מקרי הבדיקה. הסבר את המונחים בדיקות מבוססות הנחיות ובדיקות מבוססות חלוקה, לדוגמה קבוצות שקילות?**

כאשר נעשה שימוש כצפוי הרכיב שאתה בודיק אותו עושה את מה שהוא אמור לעשות.

אם יש פגמים ברכיב, לה צריכים לגלות על ידי מקרי בדיקות.

1. **מה ההבדל בין בדיקות יחידה לבדיקות של רכיבים?**

בדיקות יחידה היא תהליך של בדיקה רכיבים בודדים כל אחד בנפרד , בדיקות רגיבים היא צריגה להתמקד בתצוג של התנהגיות של הריכיבים בתאיום עם המפרת .

1. **מה החידוש שך ?TDD מה האתגר?**

מבחן מונחה פיתוח (TDD) הוא גישה לתכנת פיתוח בו

אתה לשלב את פיתוח בדיקות קוד,

  בעיקרו של דבר, לך לפתח את הקוד באופן הדרגתי, יחד עם מבחן כי

להגדיל.

1. **מתי נכון להשתמש באוטומציה בבדיקות?**

ניתן לשתמש ב באוטומציה כאשר יש בדיקות חזרות על עצםן

1. **מה מייחד בדיקות של תוכנות מדף?**

* במהלך בדיקות מערכת, רכיבים הניתנים לשימוש חוזר אשר פותחו בנפרדלסירוגין מדף מערכות עשויות להיות משולבות עם מרכיבים פתחו.המערכת המלאה נבדקת אז.
* רכיבים שפותחו על ידי חברים או קבוצות צוות שונים עשויים להיות משולביםבשלב זה. בדיקת המערכת היא קולקטיבית ולא תהליך הפרט. ב כמה חברות, בדיקות מערכת עשויות להיות כרוכות צוות בדיקה נפרד ללא מעורבות מ מעצבים ומתכנתים.

1. **מדוע אבולוציית מערכת חשובה?**

יש ארגונים השקעות ענק במערכות התוכנה שלהם הם נכסים עסקיים קריטיים.  
כדי לשמור על ערכם של נכסים אלה לעסק, הם חייבים להיות שונות ומעודכנים.  
רוב תקציב התוכנה בחברות גדולות מוקדש למשתנה ומתפתח תוכנה הקיימת במקום לפתח תוכנה חדשה.

1. **מהו התהליך הספיראלי של פיתוח ואבולוציה, אילו שלבים כולל התייחס למוצר גנרי ומוצר מותאם אישית ללקוח? מה מחדשים Rajlich and Bennett בתהליך?**

הִתפַּתְחוּת

השלב במחזור החיים של מערכת תוכנה שבה נמצא בשימוש מבצעי ומתפתח כדרישות חדשות מוצעות ומיושמות במערכת.

שֵׁרוּת

בשלב זה, התוכנה נשארת שימושית אבל רק את השינויים שנעשו הם אלה נדרשים כדי לשמור על תיקוני באגים כלומר תפעוליים ושינויים כדי לשקף שינויים בסביבת התוכנה. לא פונקציונלי החדש הוא הוסיף.

שלב מתוך

התוכנה עדיין עשויה לשמש אבל אין שינויים נוספים נעשים את זה.

1. **תאר את תהליך האבולוציה, מהם השלבים?**תהליכי האבולוציה תוכנה תלויים ב:  
   - הסוג של תוכנה מתוחזקת.  
   - תהליכי הפיתוח המשמשים.  
   - כישורים והניסיון של האנשים המעורבים.
2. **במה שונה התהליך בסעיף 06 מתהליך פיתוח תוכנה כללי?**

הבדל קריטי הוא שהשלב הראשון של שינוי יישום עשוי להיות כרוך בהבנת תכנית, במיוחד אם מפתחי המערכת המקוריים אינם אחראים ליישום השינוי.  
בשלב הבנת תכנית, אתה צריך להבין איך התכנית בנויה, איך זה מספק פונקציונאלי וכיצד השינוי המוצע עשוי להשפיע על התכנית.

1. **מדועג למערכות מותאמות custom העלויות של שלב האבולוציה עולות פעמים רבות על עלות פיתוח התוכנה?**

פיתוח תוכנה ואבולוציה יכולים להיחשב כתהליך משולב, איטרטיבי שיכול להיות מיוצג באמצעות מודל ספירלה.

למערכות מותאמות אישית, עלויות תחזוקת תוכנה בדרך כלל עולה על עלויות פיתוח תוכנה.

התהליך של אבולוציה תוכנה הוא מונע על ידי בקשות לשינויים וכולל ניתוח שינוי השפעה, תכנון שחרור ושינוי יישום.

החוקים של ליהמן, כמו רעיון שהשינוי הוא רציף, מתארים מספר תובנות נגזרות ממחקרים ארוך טווח של התפתחות מערכת.

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

שינויים דחופים עשויים להיות מיושמים מבלי לעבוד דרך כל השלבים של תהליך הנדסת תוכנה. אם תקלה במערכת רצינית יש לתיקון כדי לאפשר פעולה רגילה להמשיך.  
אם יש שינויים עסקיים שדורשים תגובה מהירה מאוד ( למשל שחרורו של מוצר מתחרה).

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

שיטות AGILE מבוססות על פיתוח מצטבר כך שהמעבר מפיתוח לאבולוציה הוא אחד חלק.  
אבולוציה היא פשוט המשך תהליך הפיתוח המבוסס על גרסאות מערכת תכופות.  
בדיקות רגרסיה אוטומטיות היא יקרות במיוחד כאשר נעשים שינויים במערכת.  
צוות האבולוציה יצטרך להתחיל מהתחלה פיתוח בדיקות אוטומטיות והקוד במערכת שאולי לא היה REFACTORED ופשוט כצפוי בפיתוח זריז.

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

* שינוי מתמשך
* מורכבות גוברים
* אבולוציה תכנית גדולה
* יציבות ארגונית
* שימור של היכרות
* צמיחה מתמשכת
* איכות יורדת
* מערכת משוב

הם לא יתקיימו ב:

* הפסיכולוג עטוף מוצרי תוכנה.
* מערכות המשלבות מספר רב של רכיבים ממדף.
* ארגונים קטנים.
* מערכות בגודל בינונית.

1. **מהם שלושת סוגי תחזוקת התוכנה? מדוע קשה לעיתים להפריד ביניהם?**  
   שלושה סוגי תחזוקה הם:

* אחזקה לתיקון ליקויי תוכנה.
* תחזוקה להתאים תוכנה לסביבת הפעלה שונה.
* תחזוקה להוסיף או לשנות את הפונקציונאליות **של המערכת.**

קשה להפריד ביניהם כי מוצרי תוכנה כלליים הם אמרו להתפתח כדי ליצור גרסאות חדשות.

1. **מהו תהליך refactoring ?**  
   שיפוץ הוא התהליך של ביצוע שיפורים לתכנית להאט השפלה דרך שינוי.

אתה יכול לחשוב על שיפוץ כ'תחזוקה מונעת ', המפחיתה את הבעיות של שינוי בעתיד.

שיפוץ כרוך בשינוי תכנית לשיפור המבנה שלה, להפחית את המורכבות שלה או להפוך אותו קל יותר להבין.

כאשר אתה לארגן מחדש את תכנית, אתה לא צריך להוסיף פונקציונלי אלא להתרכז בשיפור תכנית.

1. **בהנחה כי הנך מנהל את התוכנה של הפרויקט שבקורס. הצע שיטות לאיתור הגורמים שישפיעו על אבולוציית התוכנה ?**  
   איפה צוות הפיתוח השתמש גישה זריזה אבל צוות האבולוציה אינו מכיר את השיטות זריזות ומעדיף גישה מבוססת תכנית.

צוות האבולוציה יכול לצפות תיעוד מפורט לתמיכה בהתפתחות וזה אינו מיוצר בתהליכים זריזים.

איפה גישה מבוססת תכנית כבר בשימוש לפיתוח אך צוות האבולוציה מעדיף להשתמש בשיטות זריזות. צוות האבולוציה יצטרך להתחיל מהתחלה פיתוח בדיקות אוטומטיות והקוד במערכת שאולי לא היה refactored ופשוט כצפוי בפיתוח זריז.

1. **מהם הגורמים המרכזיים המשפיעים על עלויות הנדסה לאחור reengineering ?**הגורמים המרכזיים המשפיעים על עלויות הנדסת הן :

* האיכות של התוכנה.
* תמיכת הכלי זמינה.
* היקף המרת נתונים הנדרשת.
* הזמינות של צוות המומחים.

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

אם מערכות משמשות רק מדי פעם, או על ידי מספר קטן של אנשים, ייתכן שיש להם ערך עסקי נמוך. התהליכים העסקיים הנתמכים מערכת יכולה להיות ערך עסקי נמוך אם זה מאלץ את השימוש התהליכים עסקיים יעילים.  
מהימנות מערכת אם מערכת אינה מהימנה והבעיות משפיעות ישירות על לקוחות עסקיים, למערכת יש ערך עסקי נמוך. תקופות המערכת, אם העסק תלוי התפוקות מערכת, ולאחר מכן המערכת יש ערך עסקי גבוה.

1. **מהן מערכות סוציו-טכניות, מה ההבדל בינן למערכות מבוססות מחשב )התייחס למודל השכבות(? תן דוגמה של מערכות מכל סוג.**מחשב מבוססות מערכת: הן מערכות הכוללות חומרה ותוכנה שמפותחת בצורה כך המתפעלים והמשתמשים אינם חלק מהמערכת עצמה. דוגמה למחשב מבוסס מערכת :מעבד מילים שעובד לכתוב ספרים.

מערכות סוציו-טכניות: הן מערכות הכוללות מערכות טכניות, וגם תהליכים תפעוליים ואנשים שמשתמשים ואינטראקציה עם המערכות הסוציו-טכניות נשלטות על ידי מדיניות וכללים ארגוניים. דוגמה למערכת סוציו-טכנית: מערכות פרסום לייצור ספרים.

1. **בעת עיצוב מערכת תוכנה סוציו-טכנית יש השפעה לסביבה הארגונית על התהליך- עמוד על הגורמים המרכזיים שמשפיעים**

שינויים בתהליך: למערכת בדרך כלל יש דרישות שינוי עסקיים בעקביותם צריך הדרכה ,שינויים מהותים עשויים להיות מנוגדים למשתמשים.\*\*\*\*

שינויי עבודה: שינויים במערכת יכולים להשפיע על הדרך שבה עובדים ובכך שינוי במערכת יכול להשפיע על מעמד העובדים.

שינוי ארגוניים: מערכות עשויות לשנות את מבנה הכח הפוליטי בארגון. אם ארגון תלוי במערכת זה נותן למערכת יותר כוח ושליטה.

1. **הסבר את הקשר בין מערכות סוציו-טכניות והביטוי: "מערכת היא יותר מסכום פשוט של חלקיה"**

המאפיינים של המערכת כולה קשורים במרכיבים השונים של המערכת והאינטראקציה ביניהם, כלומר המאפיינים המופקים מהמערכת כמכלול שונים מהמאפיינים המופקים מכל מרכיב בנפרד כי זה תלוי באינטראקציה בין כל מרכיבי המערכת, ולכן הם ניתנים למדידה והערכה רק כשמשלבים כל המרכיבים של המערכת ביחד, (למשל: אמינות יעילות בטיחות...). קיימים מאפיינים פונקציונליים תפקודיים שהם פונקציה הנוצרת כתוצאה מפעילות המרכיבים השונים של המערכת, ויש מאפיינים שהם פשוט דברים המאפיינים הפעילות של המערכת כולה. במילים אחרות פונקציות ומאפייני המערכת הם לא רק של מאפייני ופעילות כל מרכיב בנפרד אלא הם מאפיינים ופעילויות חדשים הנוצרים כתוצאה מהאינטראקציה בין כל המרכיבים בעת יישום המערכת ולכן המערכת היא יותר מסכום החלקים שלה כי האינטראקציה בין החלקים השנים יצרת אפיונים ופעולות שלא נמצאים בכל מרכיב בנפרד.

1. **אמינות התוכנה היא אחד המרכיבים המשמעותיים במערכות סוציו-טכניות. יש להתייחס לשלושה היבטים בהקשר זה. הסבר והדגם**

האמינות של המערכת קשורה עקרונית במרכיבי האמינות , אך לפעמים אינטראקציה בין המרכיבים השונים עלולה להשפיע על האמינות של המערכת ולשנות אותה. בגלל העצמאיות של המרכיבים השונים בתוך המודל יכולות להתרחש טעויות במערכת, כישלון המערכת יכול להיווצר כתוצאה מאינטראקציה בין המרכיבים השונים במערכת וכמעט אי אפשר לנבא את כלהאינטראקציות בין המרכיבים השונים במערכת, ולכן מדד האמינות במערכת יכול לספק תמונה מוטעית לגבי האמינות האמתית של המערכת למשל: קריסת אחד ממרכיבי החומרה או כישלון בפלט באחד ממרכיבי תוכנה ,או טעות הנעשית על יד המשתמש, כלומר קריסות הן תוצאה של כמה מרכיבים והן משתנות בדרגת החומרה.

1. **מדוע מערכות סוציו-טכניות נחשבות ללא דטרמיניסטיות ומה המשמעות?**

כלומר במערכת זו לא מקבלים אותו פלט מאותו קלט, כלומר אפילו אם הזנו אותם נתונים לא נקבל את אותה תוצאה, וזה בגלל שבמערכת הזו יש התערבות אנושית ואנשים לא מתנהגים תמיד באותה דרך כמו כן המערכת עצמה משתנה ומתפתחת.

1. **תאר את מודל הגבינה השוויצרית והקשר לגישת המערכת?**

פרוסות של גבינה שוויצרית בתוכה הרבה חורים למרות שלא כמו בגבינה החורים הזאת תמיד פותחים כיבוי, והסטת מיקומם , הנוכחות של חירום בכל מקום ב פרסות וזה לא רגיל זה גורם תוצאות שליליות , בדרך כלל זה קורה רק כמעט החורים ב הרבה שכבות לרגע בשורה כדי לאפשר מסלול של הזדמנות תאונה זאת אומרת להביא סכנה בתוך קשר מזיק עם קרבנות , החורים נובעת להגנה בגלל שני סיבות :

1)כשילון בפעולה : זאת לא פעולה בטוחה בצועה על ידי אנשים שיש להם קשר ישר עם המערכת ויש לה הרבה צורות :טעות, שגיאה קלה , שגיאות , מימוש הפרות פרוצדוראליות בקיצור כישלון בפעולה הוא ישיר ותמיד יש לה חיים קצרה השפעה על היושרה של הגנות.

2)מצב סמוי : מצב סמוי יש לה שני סוגים :\* הם יכולים לתרגם תנאים מעוררי שגיאה בתוך מקום העבודה המקומית \* הם יכולים ליצור לטווח ארוך חורים או חולשות בהגנה , מצב סימוי ניתן לזהות ותיקון לפני אירוע שלילי מתרחש. הבנה זו מובילה ליוזמת ולא ניהול סיכוני תגובות.

1. **הסבר את הפרדוקס של אמינות המערכת על פי המאמר.**

האמינות היא דינמי בגלל שהיא בטוחה ונשמרת על ידי התאמות אנושיות בזמן ואינו אירוע בגלל התוצאות מוצלחות לעתים רחוקות תשומת לב לעצמם , הפרדוקס הוא גמיש עולה חלק ממסורת צבאית ,יש ארגוני אמינות גבוהה אזרחיים ערב חלק גדול מצוות לשעבר צבאי .ארגונים צבאיים נוטה להגדיר את המטרות שלהם בדרך חד משמועתי להיות מוצלחת זה דחוף כי כל המשתתפים להבין בבירור ולשתף את השאיפות האלה. למרות ארגוני האמינות הגבוהה מצפה ולעודד שונות של פעולה אנושית תמיד עושים עבודה קשה לשמור על חשיבה עקבית של חשדנות אינטליגנטית. ארגונים של האימות הגבוהה הן דוגמאות המובהקות לגישת המערכת אינם חסינים לתופעות לוואי