



פרויקט 4 – מפרט תיכון תוכנה (SDS)

מטרת המשימה

מטרת משימה זו היא להגדיר את מפרט התכנן – Software Design Specification – של הפרויקט שלכם, מפרט שעליו יתבסס בהמשך המימוש. המפרט יבוצע כאוסף דפי ויקי מקושרים באתר הפרויקט ויתבצע סקר הנדסי על תכנן זה. בנוסף, תעדכנו את תכניות הפיתוח ותדווחו על התקדמות בנושאי סיכונים להצלחת הפרויקט ופיתוח תשתיות.

הנחיות הגשה

עד התאריך הנקוב לכל תת-משימה עדכון אתר הפרויקט ושליחת קישורים למייל הקורס נא לציין בכותרת SDS ושם הפרויקט.

הנחיות עבודה

כללי

נפגשתם ועבדתם כבר עם הלקוחות כדי לאסוף ולנתח את דרישות המוצר. המשימה הבאה היא לתכנן את המימוש ע"י בחירת ארכיטקטורה מתאימה וניתוח מונחה עצמים ראשוני של התוכנה. פעילויות אלו יעזרו להמשיך תכנון ומימוש המוצר שלכם.

כשתשלימו את מפרט התיכון ולאחריו ואת רשימת המשימות תוכלו לשכנע את הלקוחות ש:

- אתם מבינים מהו המוצר שאתם בונים
- יש לכם רעיונות מוצקים כיצד לבנות אותו
- יש לכם את המשאבים המתאימים לעשות זאת (נרחיב זאת במשימה הבאה)

המפרט נכתב כאוסף דפי ויקי במטרה להגיע לתיכון דינמי ומשותף ככל האפשר. חישבו על דפים אלו כמצגת נושמת המאפשרת לגורם חיצוני התרשמות מהתכנון שעשיתם.

תוצרי מסירה

- 1) עד להרצאה הבאה - 80% - מפרט תיכון כדפי ויקי באתר הפרויקט. המפרט יכיל דף SDS ראשי (מקושר מהעמוד הראשי) עם תוכן כדלקמן והפניות לדפים המכילים דיאגרמות תיכון מזוויות שונות. יש לכלול לפחות ארבע דיאגרמות UML מתוך הבאות לפי העניין, כמפורט בסעיפים 2-4 להלן.

כמו כן במסגרת ה- SDS עליכם לעדכן את ניהול הסיכונים בפרויקט להתחיל כבר לתכנן את הבדיקות הנדרשות מהמוצר ואת התיעוד שיימסר ללקוח (בדרך זו תוכלו בהמשך להעריך יותר נכון את המאמץ הכולל הנדרש).
- 2) יומיים אחרי ההרצאה - סקר תיכון אישי (משימה אישית 2) – כל חבר צוות בוחר פרויקט אחר ושונה מרשימת הפרויקטים ומבצע עליו סקר תיכון לפי התבנית והמחווון שבהמשך המסמך, יש להירשם כסוקר בעמוד הפגישות. מומלץ לבצע את הסקר בשיתוף עם חברי הפרויקט הנסקר וסוקרים אחרים.

הגשה: שליחה למייל הקורס של פרטי הסקר + ציון. פרסום קישור לסקר בלי ציון ברשימת הסקרים. נא לתת כותרת למייל <Project> SRS Review to: Personal 2: SRS Review to <Project>

ציון לסקר: כמשימה אישית לסוקר לפי רמת הסקר.

- 10% לפרויקט הנסקר – ממוצע הסקרים שהתקבלו. יילקחו בחשבון רק סקרים שקיבלו מעל 70 (לא סופי).
- 3) ארבעה ימים אחרי ההרצאה – 10% פרסום רשימת משימות נגזרות כתוצאה מהסקרים, עדכון מסמכי התיכון כתוצאה מהסקר (שליחת מייל עם הסבר ולאו קישור לדפי היסטוריה שבהם ניתן לראות בהערות את השינוי שהתבצע).

קישורים ללימוד והרחבה

- Amber, [Introduction to Object-Orientation and the UML](#), Especially see: [deployment](#) (and [guidelines](#)), [class](#) (and [guidelines](#)), [sequence](#) (and [guidelines](#)) diagrams, [CRC Models](#) (and another [good explanation](#)).
- Hiranabe, [Modeling in the Agile Age: What to Keep Next to Code to Scale Agile Teams](#)

בהצלחה!

מפרט תיכון תוכנה - SDS

השפעה על התהליך: מסמכי התיכון מגדירים את מוצר התוכנה שיספק את הדרישות. ההחלטות המתקבלות בהם נסמכות על הדרישות וכן על ההבנה של האפשרויות הטכנולוגיות והרכיבים הזמינים. העבודה על התרשימים השונים עוזרת לתכנן את הרכיבים השונים במערכת, הממשקים והקשרים ביניהם ולאחר מכן ניתן לעבור כבר למימוש ובדיקות.

1. דף ראשי


תיאור התוכן העיקרי של המידע
כאן ניתן לתאר את המטרות העיקריות של דפי התיכון (בסדר יורד), למשל הכנה למימוש, בדיקת נכונות, בדיקתיות, יעילות וכדו'.

2. תרשימים (ארכיטקטורת) הפצה – Deployment Diagrams


ספקו לפחות: דיאגרמת UML אחת המתארת את הרכיבים הפיסיים (כגון jar, exe, dll) שמתוכננים למוצר שלכם ואת סוג הממשקים ביניהם. את הדיאגרמות מומלץ להכין במוצר ייעודי ל-UML, לשמור את תוכנו ולייצא ממנו תמונות של דיאגרמות הניתנות להצגה בויקי. בנוסף לתרשימים יש לתאר בקצרה את הרכיבים השונים ולצרף דיון באשר לשיקולים העיקריים בבחירתם. בשלב זה אין צורך לפרט עד לרמה של כל מחלקה ואובייקט אלא להתרכז בעיקריים שנראים לכם כרגע. בפרט דיאגרמה זו צריכה להיות מפורטת וטכנית יותר מאשר זו שהגשתם במשימה ראשונה של הצעת הפרויקט.

עקיבות: לכל דיאגרמה שבחרתם לשרטט (גם בסעיפים הבאים) הסבירו במידת הצורך את הרכיבים המופיעים בה ופרטו על אלו דרישות ממפרט הדרישות היא עונה. בד"כ יש להראות שכל הדרישות מכוסות ע"י התיכון, אך אנו נסתפק בעיקריות שבהן.


3. מחלקות ותרשימי מבנה סטטי – CRC Cards & Class/Object Diagrams

 ספקו לפחות: "תרשים" עם 3-4 "כרטיסי CRC" למחלקות עיקריות שזיהיתם במוצר שלכם, לכל אחת רשימת האחריות והקשרים ביניהן. כמו כן דיאגרמת UML אחת המתארת את הקשרים העיקריים בין המחלקות. אין צורך לפרט ברמה של שדות וארגומנטים אלא בעיקר מתודות חיצוניות (=אחריות) של כל מחלקה. לצרכי המשך הקורס – חובה שלפחות מחלקה אחת תהיה אחראית על לוגיקת חישוב שאינה טריוויאלית

4. תרשימי רצף התנהגותי – Sequence Diagrams ומצב State

 ספקו לפחות: דיאגרמת רצף ו/או מצב המציגות את האינטראקציה בין חלקים שונים של המערכת תוך כדי ביצוע פונקציונאליות מסוימת.

5. שמירת נתונים \ אחסון - Persistence

 תארו כיצד נשמרים נתונים של המערכת, אלו קבצים ייווצרו ומה יהיה המבנה שלהם (ייתכן שתצטרכו תרשים מחלקות נוסף). במקרה שאתם גם מתכננים בסיס נתונים רלאציוני, ספקו תכנון ראשוני של הטבלאות והקשרים ביניהן.

6. דרישות לא-פונקציונליות

תארו בכמה משפטים כיצד התיכון שהצעתם במסמך זה עונה על דרישות כאלו.

7. ניהול סיכונים

עכשיו כשהתמונה ברורה יותר אתם נדרשים לפתח את תכנית ניהול הסיכונים. הוסיפו בעמוד הפיתוח טבלה הכוללת לכל סיכון:

- חומרת ההשפעה על הפרויקט (1-3)
- הסבירות שהסיכון יתרחש (1-3)
- אלו צעדים בוצעו להנמכת הסיכון, בעיקר ברמת התיכון
- מה תעשו אם הסיכון יתממש

לגבי הסיכון העיקרי דווחו על בדיקה מעמיקה יותר ו/או כיצד אב-טיפוס לתשתית שבניתם עוזר להתמודד עם הסיכון ומה תוצאות הבדיקה כרגע (במשימה הבאה תתבקשו להעלות את הקוד לאתר הפרויקט) – יש לשים הפניה לדיווח מהעמוד הראשי בוויקי. יש לכלול גם קישור למדריך בסיסי שלפיו אתם לומדים הסביבה המתוכננת – למשל מרשימה זו <http://www.playframework.com/documentation/2.2.x/Samples>.

8. תיעוד המוצר

יש להתחיל לפתח את התיעוד למשתמש בוויקי לפחות ברמה של ראשי פרקים.

9. תוכנית בדיקות (ראשונית)



תארו אלו מאפיינים של המערכת מתוכננים להיבדק ומדוע זה מספיק. פרטו כיצד יעשו בדיקות אלו. תארו בדיקות יחידה (עבור רכיבים בודדים אך משמעותיים במערכת), בדיקות מערכתיות (אינטגרציה) ובדיקות שמישות (usability). כמו כן נסו לקשר את הבדיקות ככל האפשר לדרישות המופיעות במסמך ה-SRS. תארו כיצד תעקבו אחרי תקלות המתגלות תוך כדי שימוש ובדיקות.

הערה: מכיוון שנושא הבדיקות טרם נלמד, עשו כמיטב יכולתכם ותכננו לעדכן דף זו באיטרציה שבהמשך הפיתוח. הנה דוגמא אפשרית למבנה הדף:

1. תכנון בדיקות יחידה (לרכיבים משמעותיים)
- א. מהי הבדיקה (מטרה וכיסוי)
- ב. אופן פיתוח הבדיקה ועל ידי מי
- ג. תדירות שימוש בבדיקה
2. תכנון בדיקות מערכתיות
- א. כנ"ל
3. תכנון בדיקות שמישות \ חווית משתמש
4. כיצד תעקבו אחרי תקלות שייתגלו

סיכום סקר תיכון תוכנה - SDS

הסבר והוראות לסוקר:

סקר התיכון משמש כמה מטרות:

1. לוודא שהפרויקט ניגש לפתור את הבעיה שלפניו בצורה סבירה ומתקדם לעבר ההצלחה. אמנם ייתכן שגם המנהלים של חברת סבב"ה יסקרו את המפרט וינסו לתת משוב, אך תמיד כדאי לשמוע דעות נוספות.

2. הזדמנות לחשיבה ביקורתית ונתינת משוב בונה (יכולות שתזדקקו להם בהמשך הדרך בקריירה של מהנדסי תוכנה).

מחוו: הניקוד ינתן לפי עמידה בדרישות המשימה שהוגדרו לעיל לפי הסעיפים הבאים כדלקמן:

- 15% - כללי: ארגון ה-SDS, האם קריא וברור?
- 60% - תרשימי מחלקה – כ-15% לכל תרשים, לכל אחד בדוק: האם התרשים ברור? האם פשוט? האם הוא מקושר לדרישות? האם הוא מביא לידי ביטוי שיקולים שונים? רמת התיכון (לפי כללים שראינו בהרצאה)
- 25% - לכל סעיף 5%: שמירת נתונים, דרישות לא-פונקצ', ניהול סיכונים, תיעוד, תכנית בדיקות ראשונית. מספיק תוכן ראשוני אך משמעותי לקבלת הנקודות.

תוכן

שם הפרויקט הנסקר	
שם הסוקר (או כינוי)	
מועד ומקום הסקר	(אם בוצעה פגישה)
משתתפים בסקר	""
ציון	(למסירה לצוות הקורס בלבד במייל)

נקודות עיקריות שעלו בסקר

1.	
2.	

סיכום קצר של הסוקר

כתוב כמה משפטי סיכום על מצב התיכון של המוצר לדעתך, אפשר להיעזר למשל בכללי התיכון הפשוט (Beck).

הצוות הנסקר: טבלת משימות הנובעות מהמפגש\סקר

נושא	פעולה נדרשת	באחריות