**תהליך: static code analysis.**

תהליך זה יכול להיות מוגדר כ-plug in דרך Jenkins שנקרא SonarQube כך שבכל הרצה של ה-build גם הפלאגין יופעל ויסרוק את הקוד שמוגדר.

אנחנו נדגים את הפעולה של הפלאגין על ידי תוכנה אחרת שעובדת אותו דבר.

מטרת התהליך היא לסרוק את הקוד ולחפש שגיאות/באגים ללא הרצה של הקוד אלא רק מסריקה שלו.

הפניה למאמר:

כל המאמר מדבר על זה.

פסקה 1 – הסבר קצר על בדיקות סטטיות

פסקה 5 – על סמך מה עובדות הבדיקות

פסקה 7 – דוגמא לבדיקות סטטיות בגאווה

סיבות לשימוש בתהליך זה:

1. זה עוזר לגלות בעיות באיכות התוכנה בשלב הפיתוח לפני שהתוכנה עוברת לשלב ה-production.
2. זה מגלה אזורים בקוד שצריכים ריפקטור או פישוט
3. מגלה שגיאות תכנות ומציג אותם
4. עוזר למתכנתים לשפר את הקוד שלהם בעתיד.

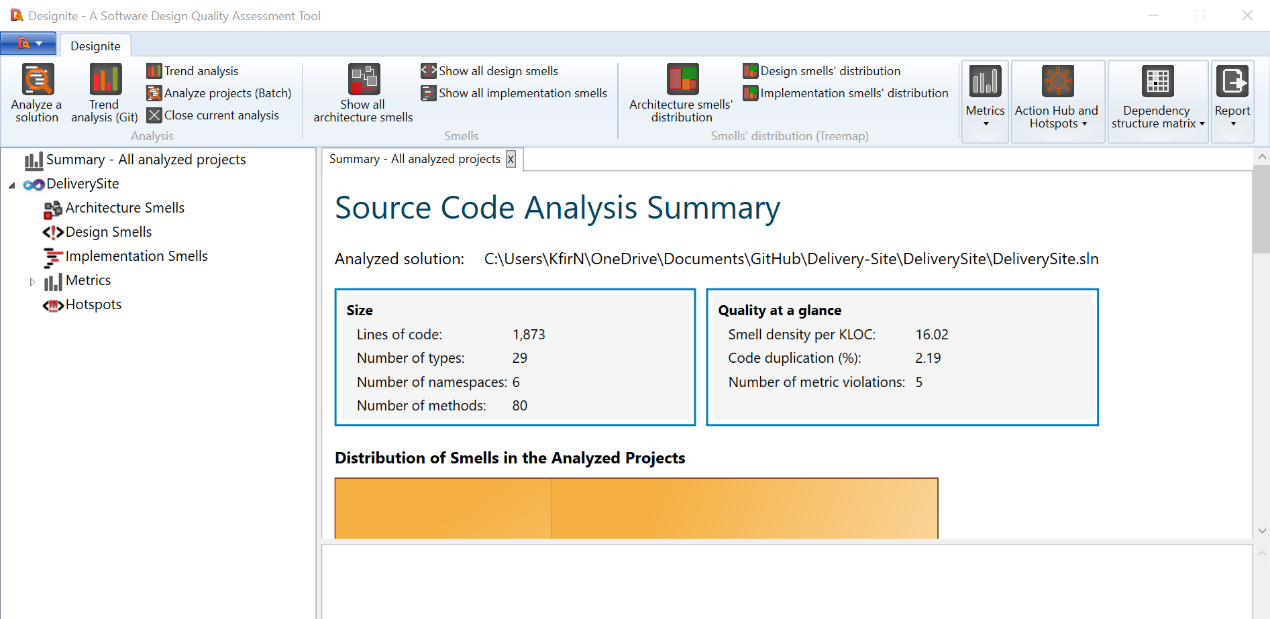
איך זה עובד?

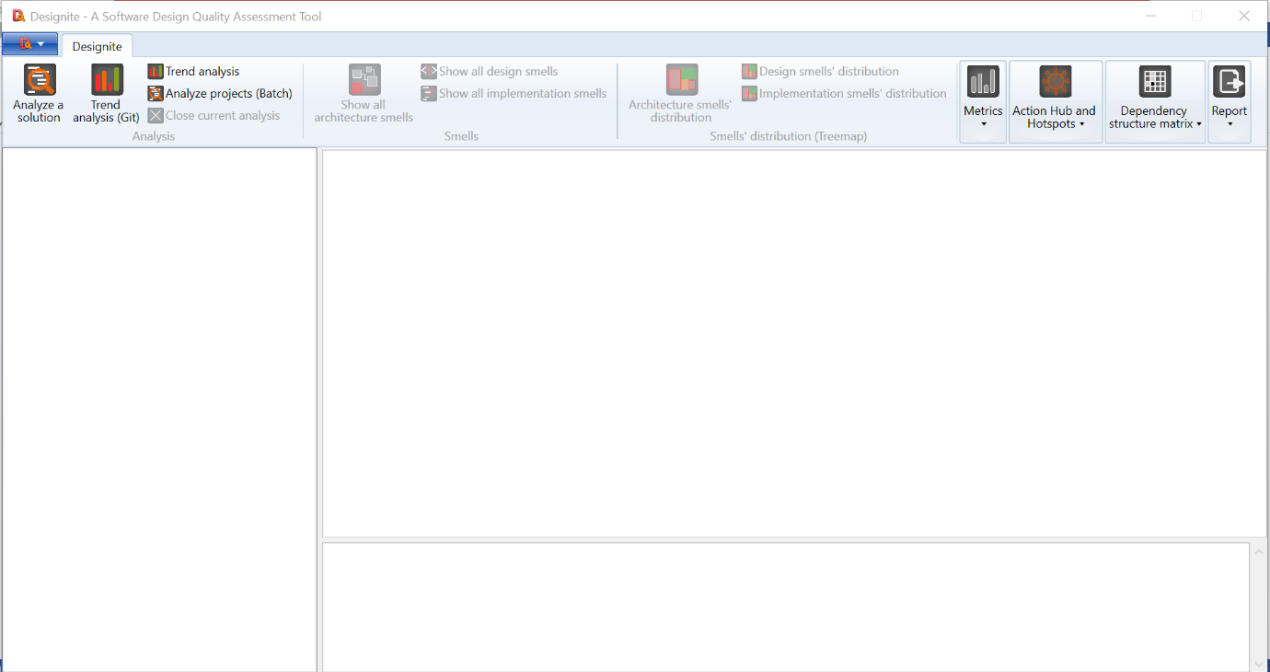
התוכנה סורקת את הקוד על מנת למצוא שגיאות.

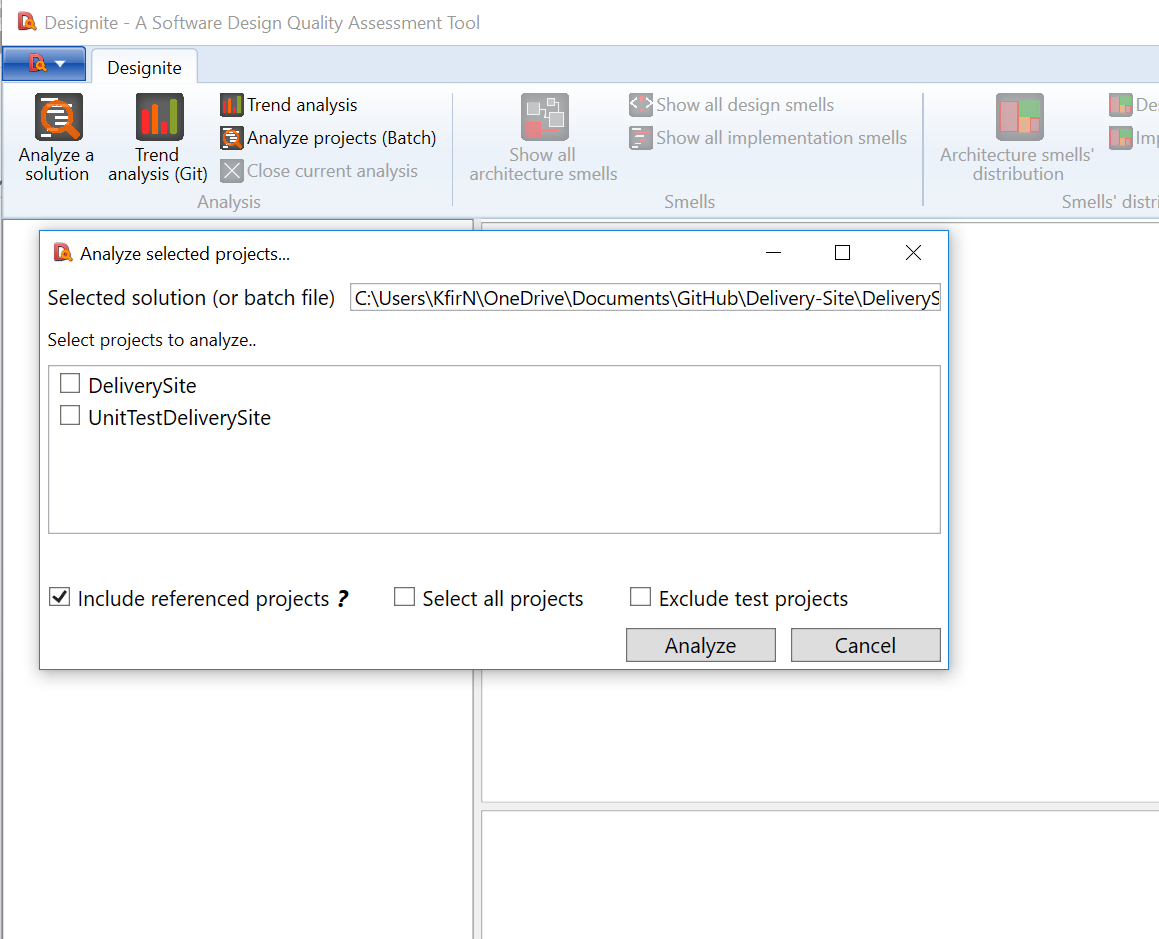
הבדיקה נעשית לפי תבניות מוגדרות כך שברגע שהתוכנה מאתרת חריגה היא מקפיצה את זה כשגיאה מתאימה.

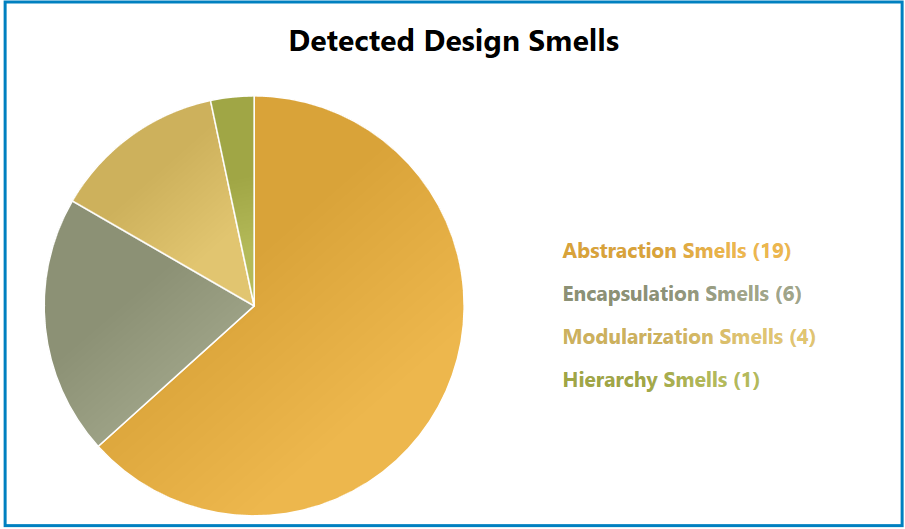
מהתוכנה ניתן לראות מגוון דברים כגון: קוד כפול, קוד מיותר, מחלקות אבסטרטיות מיותרות, מספר מחלקות, ועוד.

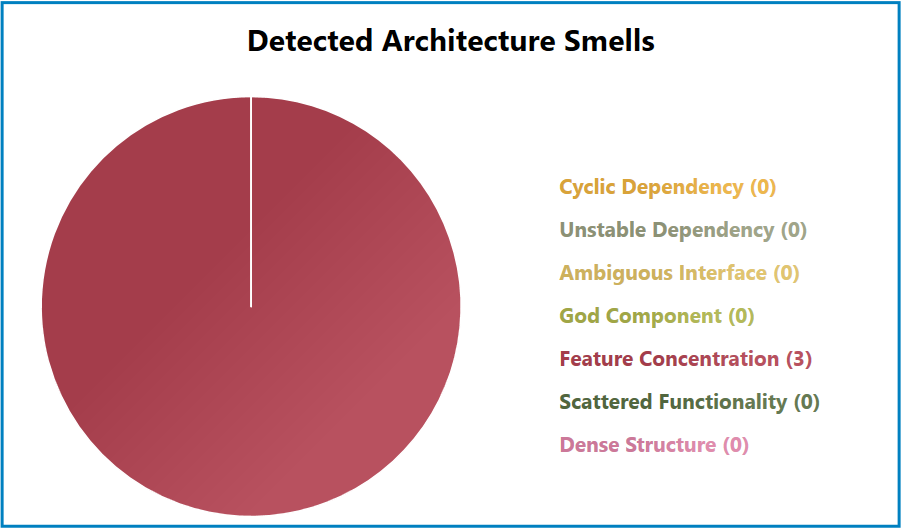
דוגמה להרצה של התוכנה:

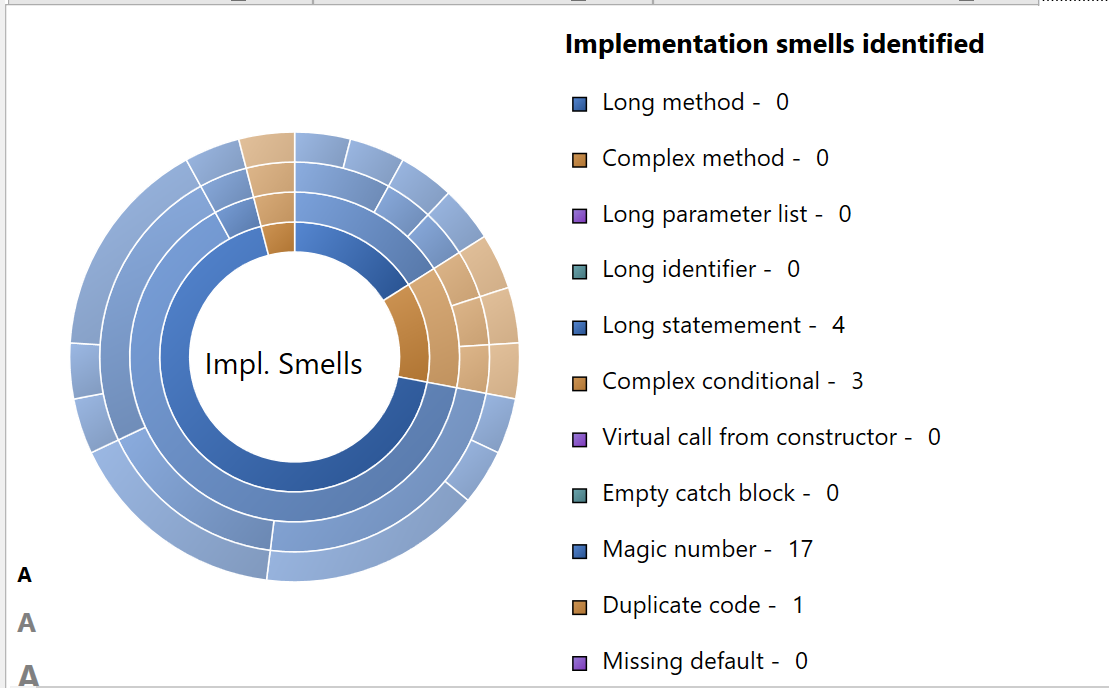












**:DevSecOps**

האתר שממנו נלקח המאמר.

<http://commons.lib.niu.edu/handle/10843/17868>

המאמר עצמו

<http://commons.lib.niu.edu/bitstream/handle/10843/17868/The%20Importance%20of%20DevSecOps.pdf?sequence=1&isAllowed=y>

תהליך זה יכול להיות מוגדר כפלאגין דרך הגנקינס כל שבכל פעם שנבצע ריצה או מהלך שלם של חבריי הצוות נבצע את הבדיקות אשר היו מוגדרות

איך זה עובד באמצעות הגניקס.:

the work with a jenkins:

page 15

One very helpful software companies can use is called Jenkins. It is a software that can help with

automation. Specifically, it is used to assist with continuous integration. With DevSecOps, there

should be frequent releases. This means there will be daily code changes and merging with

repositories. If code is not tested frequently, it could lead to a pile of defect by the time it is.

Jenkins assists with automatically testing code when it is integrated, helping to ensure that the

code contains no defects.

הסבר ומבט על devsecops:

page 10

DevSecOps

• What is it?

• Development, Security and Operations

• Weaves security into DevOps

• What is DevOps?

This is where the concept DevSecOps comes into play. It stands for Development, Security and

Operations. This is basically a guideline for how to weave security into DevOps – a practice

companies have already started using. To fully understand DevSecOps, we need to make sure we

understand DevOps.

לפני האבטחה (כשלים שהיו בעבר) וסיבות מדוע חשבו על זה. devops-before security:

page 11

DevOps – Before Security

• Restructures app lifecycle

• Core Principles:

• Collaboration and communication

• Agility towards change

• Software design

• Failing fast and early

• Innovation and continuous learning

• Automating processes and tools

• However, cycle lacks security

שילוב הקוד ואיך הוא אמור לעבוד ולהתפעל

Integration

• Developers need to learn security

• Add security checks for each small code release

• Use tools to both monitor and identify security issues

Clearly this concept is something beneficial for companies to use, but with any new

methodology, a company would need to figure out how to implement it into their work flow.

With DevSecOps, this means changing the development process from start to finish so that it

incorporates security. So, first off developers need to learn about security.

המטרה הסופית ההסבר על המשמעות המערכת והפיתוח שלה ובמה הוא תורם:

page 20

Final Thoughts

• Obviously, security is important

• DevSecOps offers the framework to bake security into an

application

• There are already many tools available

With all of that said, here are a few final thoughts. Obviously, security is important for the fact

that technology is constantly being developed and it is only being released faster. Given how

many security breaches there have been, the need for secure applications is only increasing. So,

thoughts of security need to be generated towards to the start of development.

DevSecOps is a framework with the goal to bake security into an application. It makes it so that

security is everyone's responsibility, not just the security team. And there are already many tools

available to assist with the integration of DevSecOps. Though it seems like a simple concept, it is

something that will help increase the overall application security and would help prevent data

breaches.

•Bibliography

Armerding, Taylor. “The 17 Biggest Data Breaches of the 21st Century.” CSO Online, CSO, 26

Jan. 2018, www.csoonline.com/article/2130877/data-breach/t5he-biggest-data-breachesof-

the-21st-century.html.

Bearak, Steven. “2018 Data Breaches - The Worst Breaches, So Far.” IdentityForce, 1 May

2018, www.identityforce.com/blog/2018-data-breaches.

Casey, Kevin. “Why DevSecOps matters to IT leaders.” The Enterprisers Project, Red Hat, Inc.,

15 Jan. 2018, enterprisersproject.com/article/2018/1/why-devsecops-matters-it-leaders

סיבות לשימוש בתהליך זה:

1.זה מונע פרצות אבטחה מיותרות

2.מאפשרת לתת אבטחה דבר שיאפשר להגביר את האבטחה הכוללת של הישום

3.למנוע נתונים והפרות

**דרישות**

הצעד הראשון בפרויקט תוכנה הוא בדרך כלל מסמך הדרישות. באופן מסורתי היא מטפלת בפונקציונליות, אך היא צריכה גם לטפל בבעיות אבטחה. זה צריך להעריך את רמת הסיכון הקשורים לפרויקט כולו ולזהות את הדאגות המיוחדות שלו. השאלות בהן הוא מתייחס כוללות מי ישתמש בתוכנה, כיצד משתמשים או תהליכים לקבל גישה, וכיצד המוצר יהיה אינטראקציה עם תוכנות אחרות. כל אחת משאלות אלה מרמזת על בעיות בניהול סיכונים.

הלקוח צריך להיות מעורב בתהליך צריך לספק קלט על מה צריך את ההגנה הגדולה ביותר. מוצר מוצלח חייב להיות גם שמיש ומאובטח, ומסמך הדרישות צריך לספק הדרכה ליצירת מוצר שלא להקריב אחד לשני.

## פריסה

התוכנה מאובטחת רק אם היא מוגדרת בצורה מאובטחת סביבת השרתים, יצירת משתמשים מורשים, פריסת מפתחות גישה והחשבון שבו פועל הקוד הם רק חלק מההיבטים המשפיעים על בטיחות הקוד. אנשי תפעול צריכים להבין את הגורמים הללו ולעשות רשימת תיוג של נושאים קריטיים.

התוכנה שתפרוס צריכה לעבור סדרה סופית של בדיקות אבטחה, והיא תוחזר מיד אם היא תיכשל בכל אחת מהן.

## פעולות

לאחר פריסת התוכנה, כלי ניטור הרשת צריכים לעקוב אחר התנהגותה ולדווח על חריגות משמעותיות כל סימן לכך שהתקפה צברה דריסת רגל דורשת תיקון מיידי. אוטומציה היא בעלת ערך כמו שמירה על יישומים כמו בבניית אותם.

תוכנות של צד שלישי שעליהן מותקנת התוכנה מותקנת גם תשומת לב. צוות הפעולות צריך לעקוב אחר דיווחים על פגיעויות שהתגלו ותיקונים עבורם.

בדיקות חדירה הוא בעל ערך לגלות כל חולשות לפני התוקף עושה. זה צריך להיות מתואם עם אופ, כך שזה לא להפריע לשימוש רגיל של התוכנה או נזק נתונים חיים. זה לא תמיד אומר כי אופס צריך לדעת מראש כי זה מבחן; התגובה שלהם עשויה להיות חלק ממה שנבדק.

## עדכון מחזור

כל עדכון זקוק לטיפול זהה לשחרור הראשוני, גם אם הוא מתרחש תחת לחץ. עדכונים מצטברים תכופים הם טובים, אבל רק אם הם לא מציגים סיכונים חדשים. עדיף לחזור אחורה לגרסה טובה של קוד מאשר לדחוף גרסה חדשה ללא בדיקה נאותה.

עדכונים צריכים לקחת בחשבון את מצב קוד של צד שלישי אשר נכנס לבנות. זה אולי צריך להיות מוחלף על ידי גרסה חדשה יותר להתמודד עם הבעיות שהתגלו לאחרונה. תוכנה מסחרית חופשית זמינה כדי לסייע במעקב אחר התלות של יישום ולשמור אותם מעודכנים לפי הצורך.

## סיכום

המעבר ל- DevSecOps דורש כמות גדולה של חשיבה מחודשת על תהליכי הפיתוח, על ידי ההנהלה ומהנדסים. אם צוות כבר משתמש DevOps, זה לפחות חצי הדרך שם. DevSecOps פשוט לוקח את הצעד ההגיוני הבא של הבאת הביטחון לתוך שיתוף פעולה הדוק עם עיצוב תוכנה, פיתוח ותחזוקה. ביצוע המעבר יגרום לקוד עם פחות פגיעויות, כלומר פחות שיבושים ומצבי חירום.