12 באוגוסט, 2018

# הנדסת שילוב אדם מערכת (שא"מ) – הגדרת הדיסציפלינה

## טיוטא להערות

כתב: אבי הראל

### כללי

בפיתוח מערכות מורכבות נדרשת מומחיות של מהנדסי מערכות ושל מהנדסי קוגניציה. השילוב בין שני סוגי המומחיות הללו הוא לעתים בעייתי. את הכשל של מערכות רבות, שמתבטא לעתים באסונות, ניתן לייחס לכשל בשילוב זה. הדוגמאות לכך רבות מספור, בתחום התחבורה, התעשיה הכימית, תעשיית האנרגיה, ציוד ביתי, מערכות מידע, תוכנות משרדיות ומערכות תקשורת. קיים צורך לנתח את גורמי הכשל ולהציג דרך להמנע מהם.

### הצורך בדיסציפלינה חדשה – בין הכסאות

כ-10% מהפעולות של מפעילי מערכות ותהליכים הן בטעות. למרבית הפעולות השגויות המפעיל אינו מודע כלל, ולכן תגובה המערכת אליהן היא בלתי צפויה. כמחצית מזמן התפעול מתבזבז על הבנת ההתנהגות הבלתי צפויה של המוצר. היחס בין זמן התפעול המוצלח לבין הזמן המבוזבז הוא כ-1:1, בסדרי גודל נמוך יותר הערכים שנחשבים לסבירים עבור היחס MTBF ל-MTTR.

באופן טיפוסי הבעיה היא שמהנדס המערכת מסתמך על הנחות שגויות לגבי יכולת המשתמשים והמפעילים, ומהנדס הקוגניציה אינו מכיר את פרטי התכן, ואינו מודע לשגיאות שעושה מהנדס המערכת. להלן מספר דוגמאות:

#### איכות ההתרעות

בטיסה 296 של חברת אייר פראנס, שנערכה בשנת 1988 במסגרת מפגן אוירי , המטוס לא הגיב לפקודת נסיקה והתרסק ( http://en.wikipedia.org/wiki/Air_France_Flight_296 ). בניתוח הסיבות לתקלה דווח על שני פגמים באמינות מערכת הבקרה:

- OEB 19/1 – פגם בהאצת מנועים במצבים של טיסה בגבה נמוך
- OEB 06/2 – פגם במדידת גובה המטוס.

פגם שלישי, עליו הדוח לא הצביע, היה בשיטת הדיווח לטייס לגבי מצב המערכת. בשניות שלפני ההתרסקות, המנועים פעלו במוד של סיבובי סרק. במצב זה, המנועים לא הגיבו לפקודת הטייס. המטוס היה במצב חריג, אבל הטייס לא היה מודע למצב החריג. לו היתה מערכת הבקרה מתוכננת על פי עקרונות האמינות התפעולית, היא היתה מתריעה לטייס על המצב החריג, והטייס היה יכול להגיב בזמן.

זהו מקרה קלאסי של נפילה בין הכסאות. מהנדס המערכת מצפה ממומחי השימושיות לתכנן את ההתרעות תוך התחשבות בגורמי אנוש, אבל נכשל בהגדרת כל מצבי המערכת עליהם צריך להתריע.

## תפעול במצבי תקלה

סופת ברקים ב-13 ביולי 1977 גרמה לתקלות במספר תחנות כוח, וכתוצאה מכך לעומס יתר בתחנות כוח אחרות באזור ניו-ג'רסי וניו-יורק. במשך שעה, המפעיל של מרכז הבקרה של קון אדיסון לא מצא כיצד להפעיל את התוכנה שמאפשרת לפזר את העומס בין תחנות הכוח, עד שלבסוף כל המערכת קרסה והעיר ניו-יורק שקעה בעלטה (http://en.wikipedia.org/wiki/New_York_City_Blackout_of_1977) .

זוהי דוגמא נוספת של נפילה בין הכסאות, כאשר מהנדסי המערכת דואגים לכך שהמערכת תאפשר להתאושש מתקלות, מהנדסי התוכנה מפתחים את ממשק ההפעלה שמאפשר זאת, אבל אינם מתחשבים במגבלת המפעיל הבלתי מיומן.

## איכות המידע של התרעות במצבי חירום

ב-28 במרץ 1979, הליבה של יחידה 2 בתחנת הכוח הגרעיני "אי שלשת המיליון" בפנסילבניה הותכה. תהליך איתור התקלה נמשך חמישה ימים. הקושי באיתור התקלה נבע מריבוי התרעות בלתי ממוקדות, בלתי רלבנטיות, שגויות ומטעות.

בתהליכי התכנון המסורתיים, מהנדסי המערכת היו אלו שתכננו את מערכת ההתרעות. הכלל היה פשוט: לכל אינדיקציה למצב חריג, מתריעים. הבעיה היתה שההתרעות לא הצביעו על מקור הבעיה. בעקבות ארוע זה הוכנסו מהנדסי גורמי אנוש לתהליכי התכנון של תחנות כוח, והוגדרו תקנים שיבטיחו את יעילות תהליכי איתור תקלות.

## אמינות ההתרעות לגבי תקלות

דוגמא זו מוכרת היטב למרבית בעלי הרכב. הדוגמא היא של התחממות מנוע הרכב במצב של חוסר נוזל קירור. במקרים כאלו, האינדיקציה לגבי חום מנוע מטעה, מכיוון שמד החום מודד את חום נוזל הקירור, והמדידה אינה אמינה במצב של חוסר בנוזל הקירור. מדי יום, מאות מנועים ניזקים באלפי שקלים בגלל הטעיה זו.

בתהליכי התכנון המסורתיים, מהנדסי גורמי אנוש משולבים בתהליכי תיכון כולל עיצוב ההגה, ידיות הפיקוד, המושבים, לוח המחוונים ועוד. באופן מסורתי, מהנדסי גורמי אנוש אינם נחשפים ללוגיקה של ההתרעות, ולכן טועים בהבנת משמעות ההתרעות לגבי מהות התקלות בכלי הרכב.

## אמינות המידע לגבי מצב ההתרעה

מערכת שהורכבה ממצלמות אינפרא-אדום וממרכז בקרה שימשה לאיתור חדירות למתקנים בטחוניים. המצלמות תוכננו כך שאיפשרו זיהוי תנועה ושיערוך של המרחק לנקודות חקירה. המחשב במרכז הבקרה התריעה קולית בכל מקרה של גילוי תנועה. המערכת נבחנה ואושרה לשימוש בבדיקת שימושיות קלאסית. במספר תרגילי בדיקת עירנות, התברר שמפעילי המערכת מגיבים לאט מדי, או שאינם מגיבים כלל. הבעיה אובחנה כבעיית איכות תפעולית. הסיבה למחדלי המפעילים היתה ריבוי התרעות לגבי תנועת בעלי חיים, כלי רכב שנעו באזור המתחם ושיחים שנעו ברוח. מהנדסי המערכת לא היו מודעים לצורך למנוע התרעות שווא, ומהנדסי השימושיות לא היו מודעים למצבים הבעייתיים שגורמים להתרעות הללו. גם כאן, האיכות התפעולית נפלה בין הכסאות.

## מניעת טעויות מפעיל

לא מעט מנויים של חברת הכבלים מתקשים להשתלט על השלט ולצפות בטלוויזיה בגלל ההפעלה הדו-שלבית של השלט - ראשית עליהם להדליק את הטלוויזיה ולאחר מכן הם צריכים להפעיל את הממיר ולזפזפ בו לערוץ המבוקש. צופים רבים נתקעים בדרך, למרבה מבוכתם. שלט הממיר הדיגיטלי תוכנן כך שניתן להפעיל באמצעותו גם את הממיר וגם את מקלט הטלוויזיה על מנת לחסוך שימוש בשני שלטים. זאת, במטרה לאחד את השלט של הממיר עם שלט הטלוויזיה ולחסוך מהמשתמשים, לפחות באופן חלקי, את הרדיפה הבלתי פוסקת אחר אחד השלטים שאבד בנבכי הסלון. לשם כך מעצבי השלטים הוסיפו לשלט של הממיר את המקשים השימושיים ביותר של שלט הטלוויזיה, כולל כיבוי והפעלה, שליטה בעוצמת הקול, סקירה ובחירת תחנה. על מנת להמנע מהגדלת מימדי השלט, המפתחים הוסיפו לו שני מקשים חדשים, שמאפשרים מעבר בין מצב שליטה בטלוויזיה לבין מצב שליטה בממיר. ההנחה היתה כי למשתמש נוח להשתמש באותם מקשים כאשר מדובר בפונקציות דומות (כפתורי הערוצים משמשים גם למעבר ערוצים בטלוויזיה וגם בממיר הדיגיטלי), "כי הוא כבר רגיל אליהם". הנחה זו נראית סבירה והגיונית, ומשתמשים רבים מוכנים ללמוד את דרך הפעולה, ומפעילים את המערכת ללא קושי. משתמשים רבים אחרים, לעומת זאת, מתקשים לעקוב אחר מצב המערכת. כתוצאה מכך, הם טועים בכך שהם מכבים את הממיר הספרתי במקום את הטלביזיה, ובכך שבניסיון לשנות ערוץ בממיר הם מסיטים את הטלביזיה מערוץ הקליטה. המשמעות הפיננסית של בעיית השלט היא עלייה בלתי סבירה בהוצאות התפעול של חברות הטלוויזיה וירידה במכירות כתוצאה מחוסר שביעות רצון הלקוחות.

גם בדוגמא זו, מהנדסי גורמי אנוש היו שותפים בעיצוב הצורה, המקשים והתצוגות בשלט רחוק, אבל לא היו שותפים בהגדרת לוגיקת ההפעלה. בהנדסת גורמי אנוש קלאסית הדגש הוא על התמצאות בתהליכי הפעלה במצבים נורמליים, והדעת אינה ניתנת במידה מספקת למצבים של טעויות תפעול. גם בדוגמא זו, האיכות התפעולית נפלה בין הכסאות.

## משמעות לתפוקה של תהליכי יצור

מערכת יצור מבוקרת GUI מאפשרת למפעיל לשלוט במספר פרמטרים של מספר קווי יצור. התיכנון התבסס על ארכיטקטורת SOA, כאשר מכונות ומרכיבים אחרים מיוצגים על ידי אובייקטים, הפרמטרים מיוצגים על ידי התכונות של האובייקטים והפעולות שמתבצעות על המכונות מיוצגות על ידי שיטות שמגדירות את השירותים. ממשק המשתמש כלל מסך יעודי לכל אובייקט, שאיפשר למפעיל להציב ערכים לכל התכונות ולהפעיל את כל השירותים.

מפעילים רבים סברו שהתכנון הוא לוגי. למרות זאת, קרה לעתים קרובות שכששינו את קו היצור, הם שכחו להציב את כל הפרמטרים הרלבנטים לקו היצור החדש. לפיכך, כדי להבטיח שכל הפרמטרים הוצבו כנדרש, בכל שינוי של קו היצור, הם נאלצו לפני תחילת היצור לבדוק את תקינות התהליך בדרך של ניסוי. תהליך זה ייקר את עלות היצור וגרם להפסדים. בהמשך, המפעילים הגדירו טופס פרמטרים של כל תהליך יצור, ובגירסת השידרוג, הם מימשו את הטופס בתוכנה, וזהוזילו בכך את עלויות היצור.

מהנדסי השימושיות עזרו בעיצוב מסכי ממשקי ההפעלה, וכן ביצעו ניסויי שימושיות עבור תהליכי יצור עיקריים. בדיקת השימושיות של המעברים בין תהליכי היצור היתה יכולה ללא ספק לאתר מראש את המצבים של טעויות מפעיל בהצבת הפרמטרים, ולמנוע את ההפסדים בגירסא הראשונה. גם בדוגמא זו, ההפסדים נבעו מכך שהאיכות התפעולית נפלה בין הכסאות.

## ניתוח הצורך

באופן מסורתי, מהנדסי גורמי אנוש אינם נחשפים למידע על מצבי ההפעלה החריגים. מפתחי מערכות אינם מודעים בדרך כלל לסכנות הכרוכות בתפעול במצבים חריגים, ואינם מצביעים בפני מהנדסי גורמי אנוש על מגבלות השימוש בתרחישים השונים. הם אינם מתודרכים לתכנן את מניעתם, ומהנדסי גורמי אנוש אינם מתודרכים להתריע עליהם.

בעקרון, קושי זה קשור מכך שהנושא של גורמי אנוש בהנדסת מערכות נמצא עדיין ברמה של הצהרת כוונות, והוא חסר עדיין ברמה הפרקטית. עדיין אין מסורת ומתודה סדורה של הגדרת דרישות וקריטריונים לאיכות בשימוש ובתפעול, שצריכות לכוון את מהנדסי הקוגניציה. באופן טיפוסי, מפרטי המערכות אינם כוללים הנחיות בנוגע למדדי ביצוע שונים, ולשימוש בהם להגדרת קריטריונים לאבטחת איכות התפעול. הקריטריונים להם הנדסי הקוגניציה נדרשים צריכים להתייחס למהירות ביצוע ולרמת הטעויות המותרת בשלבים השונים של רכישת מיומנות בתפעול. זאת, כאשר הדרישה למהירות ביצוע נוגדת את הדרישה למניעת טעויות. כמו כן, במפרטי המערכת חסרה בדרך כלל התייחסות להתאמה לתפקידי המפעיל השונים: הזנת נתונים, בקרה, תחזוקה, בדיקות וכיו"ב.

קושי נוסף קשור באפיון והשיטות המקובלות לניתוח גורמי הכשל ולמניעתם. באופן מסורתי, הניתוח מתייחס לאירועי כשל צפויים, ברמת הטריגר. הניתוח מתבצע בשיטות של עץ תקלות, HAZUP, FMEA וכיו"ב. המניעה היא בדרך של אבטחת איכות ברמת הרכיבים ובדיקות ברמת המערכת. הנדסת מערכת אינה עוסקת בנושא הבעייתי יותר, של אופני התמודדות המפעיל עם אירועי כשל. בין היתר, הנדסת מערכות עוסקת גם בנושאים של חסינות מערכות, אבל השיטות לאבטחת חסינות מתמקדות באירועי כשל פשוטים וצפויים. הנדסת מערכות מסורתית אינה מתמודדת עם מצבים שנחשבים לבלתי צפויים, בגין שילוב של מספר גורמי כשל שסבירותם נמוכה, ועם בעיות של התמצאות המפעיל במצבים הללו.

בעיה נוספת נובעת מכך שהנדסת קוגניציה עדיין לא הגיעה לבשלות. דיסציפלינה זו מבוססת על בסיס ידע נרחב של מדעי הקוגניציה, אבל ידע זה עדיין אינו מתורגם לכללים הנדסיים ברי תוקף. ביחוד, מורגש מחסור בקריטריונים להתאמת ממצאי מחקר במדעי הקוגניציה לסיטואציה שונות בתפעול ושימוש במערכות. דוגמא בולטת למחסור זה היא הנושא של התאמת ממשק התפעול למאפיינים של תפקיד המפעיל, כגון עומס התפקיד, על פי תכונות קוגניטיביות כגון קיבול הזכרון לטווח הקצר, על פי תיאוריות של ניהול משאבים מנטאליים וכישורים פסיכומוטוריים.

והבעיה שלשמה התכנסנו, אופן השילוב בין הדיסציפלינות, גם היא לוקה בחסר. שילוב בין הדיסציפלינות פירושו הגדרת תפקידים והקצאתם למכונה ולמפעילים, ואופן זרימת המידע בין האדם למכונה. שילוב זה חסר במודלים של הדיסציפלינות הרלבנטיות. המודל של הנדסה קוגניטיבית הוא של תפיסה, תהליכים מנטאליים וביצוע, כאשר המכונה היא בבחינת קופסא שחורה. המודל של הנדסת מערכות הוא של קלט, תהליך עיבוד ופלט, כאשר האדם נמצא בחוץ, בבחינת חידה. המפעיל אמור להבין את כוונת המתכנן, לזכור פרטים טכניים הקשורים ליישום, ולדעת ליישם את הידע בסיטואציות בלתי צפויות. שני המודלים, המערכתי והקוגניטיבי, אינם עומדים במבחן המציאות, מכיוון שהם אינם מתייחסים אל ההשפעה ההדדית במצבים חריגים.

במצב הנוכחי של חוסר בשלות בשתי הדיסציפלינות, מי שקובע למעשה את איכות השילוב זהו מהנדס התוכנה, שנדרש לעצב אינטראקציה על בסיס דרישות חלקיות ומעורפלות. התוצאה תלויה בכישורים ובנסיון של מהנדס התוכנה. האחריות לפעול על פי הגחמות (הנטיות האישיות והנסיון האישי) של מהנדס התוכנה מוטלת על המפעיל, והוא לפעמים טועה, בין בפעולה ובין בזיהוי מצב המכונה. בפרקטיקה, חסרות הגדרות של טעות, אחריות ואחריותיות.

הנטיה הטבעית של בעלי התפקידים היא לחפש את גורם הכשל במישור האישי. נטיה זו היא הרסנית בהיבט של הפקת לקחים ממצבי כשל או כמעט כשל. בהעדר הגדרות כאלו, לא ניתן למנוע את הטיית הדיון לנתיב האשמה.

## עקרונות הדיסציפלינה החדשה

בתהליך אבטחת איכות מוצרים ותהליכים, אנחנו מניחים שהמערכת תכשל בכל דרך אפשרית, ואנו מבקשים לוודא שהמערכת תשרוד את כל הכשלים, ותתאושש מהם תוך זמן סביר. באופן דומה, הנחת העבודה בתכנון ממשק הפעלה צריכה להיות שהמפעיל יעשה כל טעות אפשרית, והאתגר הוא להבטיח את תכונת השרידות וההתאוששות של המערכת.

באופן מסורתי, האחריות על הגדרת ממשקי ההפעלה של מערכות ותהליכים מוטלת על מומחים בתחום הנדסת גורמי אנוש, ובעיקר, אנשי שימושיות. הבעיה היא שמומחי שימושיות, ביחוד בעידן האינטרנט, נוטים להתמקד במאפיינים של קלות ההפעלה, ביחוד לשלבי הלימוד הראשוניים. מרבית זמנם מומחי השימושיות עוסקים בתהליכי הפעלה ראשוניים, תוך התעלמות מהבעייתיות של טעויות תפעול. כך, לדוגמא, למרות שחברת מיקרוסופט מעסיקה עשרות רבות של מומחי שימושיות בפיתוח מוצריה, כל מוצרי הקו הראשון שלה לוקים בתחום המיגון בפני כשל תפעולי. נושא ההגנה בפני כשל תפעולי נופל בין הכסאות.

## הנדסת שא"מ

מבחינים בשני סוגי כשל תפעול: טעות מפעיל, כאשר המפעיל טעה בבחירת הפקד, ותקלות מדומות, כאשר הפקד שהמפעיל בחר הוא בהתאם לכוונתו, אבל תגובת המערכת אינה תואמת את ציפיותיו. גישה זו היא בעייתית, מכיוון שבעיות קריטיות בתפעול מערכות נמצאות בתחום האפור בין הנדסת מערכת לבין הגורם האנושי. בהקשר זה, התפקיד של מהנדס שא"מ הוא לתאם בין מהנדסי המערכת, מהנדסי התוכנה ומהנדסי גורמי אנוש, בנושאים של ניתוח תקלות, תהליכי איתור תקלות והגדרת דרכים למניעת הסלמה במקרים של תקלות.

### תהליך אבטחת איכות תפעולי

בתהליך אבטחת איכות מוצרים ותהליכים, אנחנו מניחים שהמערכת תכשל בכל דרך אפשרית, ואנו מבקשים לוודא שהמערכת תשרוד את כל הכשלים, ותתאושש מהם תוך זמן סביר. באופן דומה, הנחת העבודה בתכנון ממשק הפעלה צריכה להיות שהמפעיל יעשה כל טעות אפשרית, והאתגר הוא להבטיח את תכונת השרידות וההתאוששות של המערכת.

תהליך אבטחת איכות תפעולית הוא חלק בלתי נפרד מתהליך אבטחת איכות המוצר או התהליך, עם חפיפה ניכרת בשיטות ובישומן. התהליך כולל ארבעה שלבים: תכנון, ביצוע, בדיקות ושיפור (PDCA).

המטרה של שלב התכנון באבטחת איכות התפעול דומה לזו שבאבטחת איכות המוצר או המערכת: בשני המקרים אנחנו מבקשים לוודא שכל אופני הכשל אותרו והוגדרו במפרטים, ושהמפרטים כוללים תיאור של התנהלות המערכת במקרים של כשל. האתגר של אבטחת איכות התפעול בשלב זה הוא לנבא באילו אופנים המפעיל יכשל, ולהציע דרך לוודא שאופני כשל אלו אותרו, ושהמערכת שורדת אותם.

הדרך המועדפת לטפל בכשלים היא על ידי המנעות ממצבי כשל. השיטות להמנעות ממצבי כשל כוללות אוטומציה, צמצום המערכת, פישוט ממשק ההפעלה והגדרת תהליכים על פי תרחישים. במקרים מסויימים, אין לנו ברירה אלא לאפשר למפעיל לחרוג מהפעילות השגרתית, ולטעות. זאת, כשמדובר בפעולות חיוניות של המפעיל, כגון, לצורך פתרון בעיות תפעול, במיוחד במצבים בלתי צפויים, כאשר חייבים לאפשר למפעיל שליטה מלאה במערכת. מנהל האיכות נדרש לנקוט במספר אמצעים למקרים של פעילות מפעיל שהיא בלתי צפויה, כולל פרישת רשת בטחון למניעת הסלמה, ואמצעים ללמוד להכיר את אופני הכשל, לנתח אותם ולמנוע אותם בגירסאות עתידיות.

הדרך המקובלת לבדיקות תפעול היא על ידי אבי טיפוס של ממשקי ההפעלה. לבדיקת התועלת בשלב הראשוני נהוג לבצע בדיקות שימושיות, שמתבצעות בעזרת מומחי שימושיות. בדיקות אלה אינן משתמשות לבדיקת מצבי תקלה למיניהן. בתהליכים המקובלים בתעשיה, אבי הטיפוס משמשים להדגמת אופן השגת הפונקציות העיקריות, אך לא לבדיקת תקלות תפעול או לבדיקת התפעול במקרים של תקלות מערכת. לאבטחת איכות התפעול יש צורך להרחיב את אופן השימוש באבי-טיפוס כך שהם יאפשרו יזום של מצבי תקלה צפויים ובלתי צפויים, על מנת לבחון את עמידות המערכת בפני התקלה ואת תהליך ההתאוששות.

## מתודולוגיה למניעת טעויות מפעיל

בבסיס הנדסת שא"מ נמצא מודל החסינות, המבוסס על עקרון הבקרה העצמית STAMP שפותח בעזרת מרכז גורדון להנדסת מערכות בטכניון. מודל החסינות מבוסס על האבחנה שכשל מקורו בצירוף אירועים שהסתברותם נמוכה, ושקיים קשר הדוק בין מצבי כשל לבין תפעול במצבים חריגים. מודל החסינות כולל תיאור של תגובה לאיומים שמקורם בהפרעות, שמאפשרת למערכת לסכל את האיום ולהתאושש ממנו. סיכול האיום מבוסס על זיהוי המצבים החריגים, והנחיית המפעיל כיצד לפעול בתנאים שאינם מוכרים לו. המודל כולל תיאור של תכונות המעניקות למערכת יכולת חסינות במצבים כאלו.

בבסיס המתודולוגיה לאבטחת חסינות עומד הרעיון של הגנה בשכבות המתייחסות לשלבים השונים של התפתחות האיום על פי המודל: מניעת הפרעה, תיקון אוטומטי או מבוקר, גילוי מצבי איום, התרעות למפעיל, הכוונה בתהליך איתור תקלות, תפעול במצבי חירום וכיו"ב. בנוסף, המתודולוגיה כוללת גם התייחסות לאירועים בלתי צפויים, הנשענת על רעיון הבקרה העצמית של מערכות, כאשר המפעיל נמצא בחוג הבקרה. ההנחה, שהיא ואריאציה של חוק מרפי, היא שאם התכן מאפשר למפעיל לטעות, במוקדם אם במאוחר הוא אכן יטעה, ולכן התכן צריך למנוע את האפשרות לטעות. המימוש הוא על ידי הכללה של חוקי התפעול בבסיס ידע שמשמש למעקב אחר הפעילות, ולבדיקה שהפעילות נעשית על פי חוקי התפעול. המתודולוגיה כוללת כללים ושיטות להתנהלות תקינה במקרה של מצבים חריגים, וכן ארכיטקטורה לניהול תהליך התגובה לאיומים.

## מקורות

2007 - אבטחת איכות תפעולית - בין הכסאות - פוסטר ,הכנס הלאומי לאיכות, תל אביב

2017 - שילוב אדם-מערכת- הלכה ומעשה - נייר עמדה – אפריל