

בשלבי פיתוח

מדריך לתרבות ניהול טכנולוגית בצוותים שרצים מהר

Volume 1

By Nadav Shatz CTO
& Yael Sapir-Zehavi VP HR

לפני שנתחיל | אינטרו קצר

אם יש משהו אחד ברור, זה שמחשבים לא יכולים להחליף מנהלים ומנהלות. אלף פתרונות טכנולוגיים לא יכולים להחליף את שיקול הדעת הניהולי (והאנושי) שיש למנהלים. הבחירה לנהל היא בחירה אמיצה, ומגיעה עם עוד אחריות, עוד עשייה ועוד השפעה. לא רק על המוצר - אלא גם על האנשים שאיתם אנחנו עובדים.

לכולנו יצא לעבוד עם מנהלים ומנהלות טובים יותר וטובים פחות, אבל בסופו של דבר, כולנו רוצים להיות מנהלים טובים יותר. בעוד שיש דברים שנכונים לכל סוגי הניהול, ניהול צוות טכנולוגי הוא אתגר מסוג אחר.

התחלנו לכתוב את המדריך הזה כמסמך פנימי עבורנו, וככל שהוא התעבה והשמין החלטנו לנסות ולאסוף את ה"אני מאמין" שלנו בכל מה שקשור בתרבות טכנולוגית בכלל ובפרט בתרבות ניהול של צוותים ומנהלים טכנולוגיים. אין פה נכון או לא נכון. זו לא תורה חד משמעית ואנחנו מאמינים שכל אחד יודע להחליט מה נכון עבורו ועבורה. השאיפה שלנו היא שנצליח לפתח יותר ויותר שיחות לגבי האופן שבו נכון לנהל אנשי ונשות פיתוח, ולחזק את סל הכישורים שיש למנהלים טכנולוגיים. לכן החלטנו להוציא את המדריך הפרטי שלנו החוצה.

זו הגרסה הראשונה של המדריך הזה ואנחנו בטוחים שהיא יכולה להיות יותר טובה. אחרי הכל, גם לנו יש מה ללמוד. אנחנו גם יודעים שלשחרר גרסה ראשונה זה חשוב, כי זה תמיד עדיף מגרסה שיושבת במגירה ולא רואה אור. ואנחנו נמשיך לשפר, לתקן ולהוסיף.

הסיכוי לשלמות הוא נמוך אבל כל עוד אנחנו משתפרים - אנחנו מנצחים.

אנחנו נשמח לשמוע מכם, ומאמינים שהידע שיש בחוץ תמיד יכול לעזור לכולנו להשתפר וללמוד איך לנהל טוב יותר צוותים טכנולוגיים. אז דברו איתנו - אנחנו במרחק מייל מכם.

נדב שץ, CTO | nadav@tailorbrands.com

יעל ספיר-זהבי, VP HR | yael@tailorbrands.com

2	<u>למה כתבנו את המדריך הזה?</u>
4	<u>איך תדעו שמצאתם את ה"אחד"?</u>
4	<u>התער של אוקאם</u>
6	<u>8 מנבאים להצלחה</u>
9	<u>איך הופכים אנשי פיתוח למראיינים?</u>
10	<u>The Genius Asshole - Just Say "No"</u>
11	<u>איך לתת פידבק לאנשים טכנולוגיים</u>
14	<u>9 מחשבות על ניהול אנשים טכנולוגיים</u>
20	<u>ג'וניורים: כן או לא? (בקצרה...כן!)</u>
20	<u>אז...איך מפתחים ג'וניורים?</u>
21	<u>למדו אותם ללמוד מקצוענות מהרגע הראשון</u>
27	<u>מתי יודעים שזה לא מתאים?</u>
28	<u>איך לבנות תוכנית פיתוח אמיתית לצוותים טכנולוגיים</u>
32	<u>ניהול של פרודקשן - A Must</u>
36	<u>עבודה עם סטייק הולדרס</u>
36	<u>מטרת העבודה המשותפת</u>
37	<u>עבודה עם פרודקט</u>
39	<u>הם ואנחנו</u>
40	<u>איך עושים postmortem בלי אשמה?</u>
46	<u>נספחים</u>

איך תדעו שמצאתם את ה"אחד"?

TLDR: בפרק הזה נלמד איך למצוא מאטץ' טוב כשמראיינים מועמדים לחברה שלכם, ונעבור על טיפים שמסבירים איך לעשות את זה - גם מהפן של ההתאמה הטכנית, וגם מהפן של ההתאמה האישיותית. בנוסף, נעבור על איך מכשירים מראיינים ולמה אף פעם (אבל אף פעם) לא כדאי לגייס אליכם Genius asshole.

"תחפשי את האביר על הסוס הלבן. אם מגיע רק הסוס בלי האביר - גם טוב. רק תוודאי שהסוס הוא סוס, ולא חמור"

המשפט הזה נאמר לי ע"י אמא שלי (יעל), כשחיפשתי בן זוג, אבל הוא לגמרי נכון גם בגיוסים. הרי איך אפשר לוודא שמי שפגשנו לרגע קצר הוא-הוא המאטץ' הנכון למקום העבודה שלנו?

זה לא סוד שגיוס של מפתחים טובים זו מיומנות בפני עצמה. כמנהל מגייס, אתה צריך לפתח את היכולת לגלות בזמן קצר אם המועמד או מועמדת שמגיעים לראיון יתאימו לתפקיד. וזה כמובן יותר קל כשיש מולכם "אביר על סוס לבן": מישהו שמתאים כמו כפפה ליד. יש לו את הניסיון הרלוונטי, את הוותק, הוא מכיר את הטכנולוגיה שלכם והוא תקשורתי ושותף למשימה (וגם - ספונטני ומצחיק).

זה כמובן קצת יותר מורכב, כשפוגשים מישהו שכל התכונות האלה לא מתקיימות אצלו, וצריך לפענח את הפוטנציאל שלו, ולנסות להבין אם הוא המאטץ' המתאים.

בתכלס, כמעט כל המועמדים שאנחנו פוגשים - נופלים בדיוק לקטגוריה הזו.

התער של אוקאם

מה בודקים בחלק הטכני?

אמור לי איזו שאלה אתה אוהב לשאול בראיונות ואומר לך מי אתה: נתון מילון. נתון טור מנורות. נתונה רשימת שירים. בנה פונקציה רקורסיבית שהופכת עץ בינארי מימין לשמאל ומשמאל לימין כמו שאף פעם לא תצטרך מחוץ לתואר. ממש לי רשימה מקושרת כמו שכל טכנולוגיה תתן לך בחינם. פחות אנחנו.

בחלק הטכני, אנחנו תמיד ננסה לגלות מה רמת הידע של המועמד שאנחנו פוגשים. ויותר מזה - נרצה לגלות אם חסר ידע מסוים ובעיקר ננסה להבין את צורת החשיבה שלו. אבל בשביל לבדוק את כל זה, לאו דווקא צריך לשאול שאלה טכנית.

נוכל למשל לשאול שאלות כאלו:

- + תציג לי פרויקט שאתה מתגאה בו - למה בחרת לעשות את מה שעשית?
- + אם היו לך אינספור משאבים - מה היית עושה אחרת?
- + האם זה עלה לפרודקשיון?

המטרה שלנו היא להעריך Technical proficiency על ידי זה שנגיע לפתרון בתוך מסגרת זמן סבירה שמתאימה למה שאנחנו מחפשים אצלנו בארגון. ונבדוק האם הוא נתקע בפיתרון מסויים, או מנסה להיות יצירתי. וגם כשמישהו לא עונה את התשובה הנכונה - נראה איך הוא מנסה להגיע לתשובה.

סימן טוב הוא שמישהו רואה שהפתרון שלו לא מושלם, מכיר בזה, ויודע להסביר לנו למה הוא (הפתרון) לא מושלם. גם אם הוא לא ידע ברגע הראיון איך לפתור את זה.

שאלות מצויינת שאתם יכולים לשאול בנוסף הן:

- + אתה שלם עם הפתרון שלך?
- + מה לדעתך פחות עובד בו?
- + מה היית רוצה שיהיה יותר טוב?
- + מה היית מוסיף אם היית עושה את זה שוב עכשיו? או מה היית מוריד?

עוד דרך היא לנסות לדמות סיטואציה מהיום יום שלכם כחברה. נצא מנקודת הנחה שאתם מגלמים את הדמות של מנהלי מוצר, ואתם מנסים להעביר נקודה עבור המפתח. תנו רק 80% ממנה - כמה מהר הם משלימים את מה שאתם לא מספרים להם? היכולת להבין את הפער יכולה להעיד על דרך החשיבה. וברמה המתקדמת יותר: איך משלימים את הפער? האם המועמד שואל שאלות? או בטוח בעצמו וסבור שהם יכולים להשלים את הפער לבד? ביטחון עצמי, ומהצד השני של המתרחס - ענווה, יכולה להעיד על יכולות גבוהות של מפתחים.

מה עם המובן מאליו? Correct code? אנחנו חושבים שזה הדבר שהכי קל לבחון, ויש כל כך הרבה מידע ברשת על סוגים שונים של שאלות לבחון את זה. למען האמת, אנחנו לא חושבים שיש לנו פה ערך מוסף מיוחד לתת. רק נוסף ונאמר, שאחד הערכים שלנו בחברה הוא: "Make Something Beautiful". הערך הזה אולי נשמע מאוד "מוצרי", אבל יש לו התגלמות מאוד חשובה אצלנו בצוות הפיתוח. אנחנו מאמינים שמה שיצרנו הוא יפה. לא במובן של pretty or not pretty אלא במובן של בעל ערך. משהו יפה הוא משהו שאפשר למנף אותו ליצירה נוספת ויש לו תכלית. כמעט תמיד, הדבר הזה ילך יד ביד עם פשטות. מכירים את התער של אוקאם? אז כזה. קוד יפה למשל, הוא קוד אלגנטי. יש משמעות לכתיבת קוד נקי ויפה, שהוא בעל הפוטנציאל המינימלי ביותר לגרור אחריו טעויות.

הדבר הזה דורש איזון מתמיד. החיים הם אפורים, ויש לנו גם משפט שאומר "Done is better than perfect": זה נהדר לראות אנשים Take pride בכתיבת הקוד שלהם ולא רק מונעים מלהגיע למטרה (למשל אמרות כמו - "מה זה משנה? העיקר שזה עובד"). ומהצד השני, צריך לדעת להפריד 20-80 ולבחור מתי לעצור ולוודא שהדברים עולים ורואים אור יום. מה הכי חשוב? שיקול הדעת של המפתח שמולכם. שאלה מצויינת תהיה סביב העולמות הללו: מתי תרגישו בנוח לשחרר משהו לאוויר? מה השיקולים שאתם מפעיל בקבלת ההחלטות סביב זה?

אחד הדברים המעניינים, זה שלמרות שיש פה חלוקה דיכוטומית בין החלק הטכני לאישיותי, ראיונות של אנשים טכנולוגיים מאוד שלובים זה בזה, והרבה מהסממנים האישיותיים נבחנים גם בזמן שבוחנים את החלקים הטכניים. לצורך הבהירות, הפרדנו בין השניים, אבל בראיון עצמו - אנחנו מאמינים שהם צריכים להיבחן יחד.

8 מנבאים להצלחה

אחד הדברים הטריקים בחלק האישיותי של הראיון, הוא העובדה שהתוקף החיצוני שלו כל כך נמוך. גם מנהל מנוסה, שראיין כבר המון מועמדים בחייו, מוצף בהטיות כמו אפקט ההילה (כשיש למישהו תכונה חזקה אחת, זה צובע את שאר בתכונות שלו באור חיובי יותר), אפקט הקרניים (אותו הדבר, אבל הפוך - תכונה שלילית אחת, צובעות אחרות באור שלילי). אפקט הדמיון (אנחנו אוהבים אנשים שדומים לנו, בין אם זה כי הם עשו מסלול קריירה דומה לשלנו, כי הם למדו באוניברסיטה דומה לשלנו או כי הם גרו באותה עיר שבה גדלנו ואפילו אם הם אוהדים את אותה קבוצת הספורט שלנו. אנחנו מעריכים את עצמנו כאנשים טובים וחזקים, וכשיש דמיון, זה גורם לנו לחשוב שמי שמולנו גם הוא טוב וחזק כמונו).

יש 8 תכונות שלמדנו לאורך הזמן שמנבאות בצורה הכי טובה הצלחה בתפקיד. ריכזנו אותן ביחד עם הצעות איך לבדוק כל אחת מהן:

1. תקשורתיות

היכולת לשאול שאלות ולגלות אמפתיה לנקודות מבט שונות משלך תוך כדי שיחה. תקשורתיות נבחנת הרבה בשיחה שבין מועמד למראיין. הדרך הכי טובה לבחון אותה היא כשמועמד עושה תרגיל ומשוחחים תוך כדי על מה שקורה - עד כמה המועמד משתף במה שהוא עושה? עד כמה הוא מרגיש בנוח לשאול שאלות מכוונות שהוא לא יודע משהו? עד כמה הוא יודע להסביר את הסיבות מאחורי מה שבחר לעשות (גם אם זה לא הפתרון המושלם)?

2. עבודת צוות

לא בכל חברה זה מאוד קריטי, אבל אצלנו מאמינים שזה מאוד קריטי. בארגון שיש בו הרבה עבודת צוות, ועשייה משותפת, הרצון לעזור והיכולת לבקש עזרה, היכולת להתגייס למשימה כשמישהו אחר זקוק לך והאהבה לאנשים, הם מפתח משמעותי להצלחה. שאלות בתחום הזה למשל: למה אתה אוהב לעבוד בצוות? יש לך דוגמה למקרה שבו חשבת לעצמך מזל שאתה עובד בצוות, כי התוצאה שהשגת היתה טובה יותר מאשר אם היית עובד כסוליסט? מה זה בשבילך לעבוד בצוות? מה היית מצפה מחברי הצוות שלך? איזה ערך מוסף אתה מביא איתך לצוות?

3. עצמאות

נרצה להבין איך המפתח שמולנו לומד דברים חדשים. כמה הוא עושה בעצמו, לעומת כמה הוא נעזר באחרים. תהליך הלמידה העצמית יכול להעשות בהרבה צורות, וכל אחד בוחר בדרך המתאימה לו, אבל עדיין, נרצה לוודא שהיכולת העצמאית שלו חזקה. בנוסף, נבדוק עד כמה אותו מפתח חושב לבד על ההשפעות של ההחלטות שלו, עד כמה הוא מנסה למצוא דרכים לשפר את המוצר, או אפילו את חיייהם של חבריו לצוות בשיפורים שהוא עושה על הדרך, או רעיונות שהוא מעלה. כשנדבר עם מפתח על דברים שלדעתו התנהלו פחות טוב במקום הקודם, נרצה לשאול מה הוא עצמו עשה על מנת לשפר אותם. כמובן שנשמח לשמוע שפנה למנהל שלו והציע פתרונות, או אפילו הציע לקחת אותם עליו.

4. נחישות

יש ערך גבוה ליכולת לנסות ולא לוותר, גם כשקשה, בייחוד אצל מפתחים (שנתקלים בהמון קירות). הרבה פעמים פתרון לבעיה לא מתגלה מיד, ודורש הרבה עבודת רגליים והרבה מחקר. אנחנו יכולים לראות נחישות אצל מועמדים כשהם מנסים למצוא פתרון במהלך התרגיל, ויכולים גם לשאול אותם על פתרון שהגיעו אליו אחרי דרך ארוכה של עבודה. אפשר תמיד גם לשאול על מקום שבו הם החליטו לוותר ולמה.

5. לקיחת אחריות

עד כמה מי שמולנו לוקח אחריות על הצלחות עבר וגם על דברים שלא הצליחו? ועד כמה אנחנו רואים שבתוך המשימות שלו, הוא יודע להגדיר חלקים שהוא אחראי עליהם מ-א' עד ת' שאלות בעולם התוכן הזה מתחברות לבקשה לספר על ניסיון מוצלח, וגם על ניסיון לא מוצלח, שהיה למועמד. כדאי גם לבקש שיספר על חלק שהיה אחראי עליו לחלוטין, ואיך הוא עשה את זה מהתחלה ועד הסוף.

6. פתרון קונפליקטים

עד כמה המועמד בעל גמישות? האם הוא יודע לנהל דיון בריא ונמנע מלהיכנס לפינות? עד כמה הוא מסוגל לראות את התמונה הגדולה ולקבל נקודות מבט שונות מאלו שלו? שאלות טובות בעולם הזה יכולות להיות סביב מצב שבו הוא היה ב-Disagree and commit. איך זה עבד? למה לדעתו לא קיבלו את דעתו? בדיעבד, איך הוא מסתכל על הסיטואציה היום ומנתח אותה?

7. ענווה

תכונה מאוד חשובה בחברה שבה יש הרבה מפתחים חזקים. אנחנו תמיד אומרים שאנחנו רוצים אנשים שכשהם נכנסים לחדר - הם לא בטוחים שהם האדם הכי חכם בחדר. לכולנו יש מה ללמוד, כל הזמן. ומי שאין לו ענווה - קשה לו יותר ללמוד. שאלה טובה לבחון את זה יכולה להתחבר למקום שבו המרואיין חושב שהוא לא הצליח במשהו בעבר, או טעה. כשמועמד לא מצליח למצוא דוגמא, או שכל הדברים שהוא נותן כדוגמאות קשורים בבעיות של אנשים אחרים, אנחנו יכולים להסיק שרמת הענווה שלו לא גבוהה.

8. מיקוד שליטה פנימי

מיקוד שליטה מתייחס למידת תחושת השליטה שיש לנו על האירועים שקורים לנו בחיינו. אנשים עם מיקוד שליטה פנימי מבינים שיש נסיבות חיצוניות, אבל עיקר העבודה שלהם נעשית כלפי עצמם - איפה שיש להם הזדמנות לייצר השפעה ושינוי גדולים יותר ומהירים יותר. המון מחקרים מראים שאנשים שיש להם נטייה למיקוד שליטה פנימי מצליחים יותר, מתקדמים יותר ומגיעים להישגים רבים יותר גם בחיים האישיים וגם בחיים המקצועיים (וגם מאושרים יותר - אבל זה למדריך אחר). הצד השני של אותו המטבע, הוא מיקוד שליטה חיצוני - אנשים שרואים בגורמים חיצוניים את הסיבות לדברים שקורים, ולכן מרגישים שיש להם מרחב השפעה קטן על הדברים. במצב כזה, הם נמצאים הרבה פעמים בחוויה של חוסר אונים, תסכול, מרמור ושיח מאשים. הם מרגישים שהם עצמם היו בסדר, ואם רק היו מקשיבים להם, או עושים את הדברים כפי שרצו, הכל היה מסתדר. על השאלה "מה יכולת לעשות אחרת?" הם יענו תשובות בנוסח של: "לשכנע יותר בעמדתי". זה נכון שיש הרבה אירועים שאכן לא תלויים בנו, וכל מקרה לגופו, אבל המטרה שלנו בחיפוש דוגמאות לשיחה היא לעמוד על מנגנון אישיותי, ולראות אם הוא מתאים לתפיסת האווירה שלנו, שמנסה לקדם דברים דרך היכולת שלנו להשפיע עליהם, ולא לתלות הכל באחרים. מבין כל התכונות שציינו בפרק הזה, אין ספק שלאורך הדרך למדנו שזה אחד המנבאים הכי חזקים שיכולים להיות להצלחה של אנשים בתפקיד.

הרבה חברות מאמינות שאחת הדרכים הטובות להתגבר על הטיות אלה היא להביא מספר רב של שופטים. לרוב, לשלב שני מראיינים בשלב הראשוני יכול לייצר במה מספיק טובה שתשלב מספר נקודות מבט בתוך הראיון. אפשר לייצר שכבה של מראיינים, ולשלב בין המראיינים והמועמדים השונים כדי לחוות סגנונות שונות של פרסונות מראיינות. וטיפ מאוד חשוב - קליברציה - להסתנכרן בדיוק על מה מחפשים ואיך בודקים את זה. סינכרון בין המראיינים השונים לגבי מה שמחפשים, מייצר הלימה מאוד טובה בהבנה האם המועמד שאנחנו פוגשים יכול להתאים לנו, והאם יש לו פוטנציאל להצליח איתנו.

איך הופכים אנשי פיתוח למראיינים?

בסוף, האנשים שהכי מתאימים לגייס עובדים - הם חברי הצוות עצמם. ואנחנו רוצים שכמה שיותר מהם יהיו מעורבים בתהליך הגיוס. אחרי הכל, הם יודעים הכי טוב מה בפועל צריך לתפקיד. והם גם ידעו לנבא (כמה שאפשר) את ההצלחה של המועמד בצוות.

אז איך הופכים אנשי פיתוח למראיינים? אנחנו מאמינים שחשוב להשקיע זמן בלימוד הצוות המראיין איך לראיין. החל מסדנאות, סימולציות ו-shadowing. זמן על הכביש מלמד מצויין, והלמידה הכי טובה תהיה בהיחשפות לכמה שיותר ראינות. אחרי ראיין כדאי לחשוב על סנכרון קבוצתי שבו כל אחד משקף את מה שראה, ושומע איך אחרים ראו את זה, כדי לוודא שמכילים את הצוות ואכן בוחנים את אותם הדברים (במצב כזה, דרך אגב, כדאי לתת דווקא לחדשים להביע את דעתם, כדי שלא יושפעו מדעתם של "היותר מנוסים" ויוכלו לגבש את דעתם בעצמם).

לבחור נכון את האנשים שמצטרפים לארגון זו אחת המשימות החשובות, ולא כדאי לזרוק אנשים למים, לתת להם לראיין ולקוות לטוב. אנחנו מאוד מאמינים בערך שבהשקעת זמן בהכשרת כל המראיינים. מתן הכלים הזה הוא קריטי כדי לעזור לבחירות שלהם להיות טובות ומוצלחות.

כמובן שהבחירה היא גם דו-כיוונית. וחשוב גם לדעת לספר על הארגון, להציף את אופן העשייה וללמד את המראיינים לשקף את הארגון בצורה המדויקת ביותר. מאטץ' טוב צריך להיות משני הצדדים. אין ערך לייצר תמונה של משהו שהוא לא אמיתי או שונה מאוד מהמציאות - call it like it is. לכל מכסה יש סיר, והמועמדים שלכם נמצאים שם. תהיו כנים על איך הדברים עובדים. אותנטיות תמיד מנצחת.

The Genius Asshole - Just Say "No"

כולנו מכירים טיפוס כזה. אחד שיש לו יכולות מדהימות, כישורים מקצועיים מושלמים, ניסיון מוכח ו...יחסי אנוש בעייתיים. יש כאלה שכשהם כותבים קוד - זה כמו הלחנה של יצירה, יש להם מן חוש שישי כזה, וכשהם מסתכלים על בעיה הפיתרון מבליח להם בתובנה זוהרת והם פשוט יודעים לקחת כל דבר ולשכתב אותו לרמה יותר גבוהה של אלגנטיות.

אצל ה-Genius Asshole, הערך המקצועי האדיר הזה מלווה בתחושת עליונות, או במה שניתן לתאר כ"דיווה". הם עובדים לפי כללי המשחק שלהם, הם יודעים יותר טוב מכולם, אפשר לפנות אליהם רק בצורה מסויימת והרבה פעמים, כשנעזר את האומץ לפנות אליהם לעזרה, אולי נצא עם מה שביקשנו, אבל כנראה נרגיש עם עצמנו קצת פחות טוב אחרי זה.

מניסיוננו, לא משנה כמה הערך המקצועי שלהם אדיר - הם לא שווים את זה. בכדי לבנות מחלקה מאוחדת, של צוות אחד שעובד יחד לעבר משהו גדול יותר - אין לזה מקום. זה מייצר גלים, תחושה של אי שוויון, ומשבש את האווירה בצוות. הרבה פעמים נדמה לנו שאם כבר יש לנו מישהו כזה בצוות, אנחנו לא נצליח להגיע להישגים בלעדיו. הניסיון מלמד שפעם אחר פעם, אנשים מגלים שהם יכולים להסתדר גם בלעדיהם.

ההמלצה שלנו - אל תתפשרו על הנושא הזה, אל תכניסו לצוות שלכם את מי שיריד את האחרים ולא יהיה שותף תקשורתי מלא בצוות. זה פשוט לא שווה את זה.

איך לתת פידבק לאנשים טכנולוגיים

TLDR: בפרק הזה נלמד איזה פידבק עובד הכי טוב עם אנשי פיתוח (ספוילר - כזה שהוא עכשיו ולא אחר כך), איך לתת (ולקבל) פידבק וממה כדאי להיזהר.

מה חמוץ? לשמוע מהמנהל שלך שהוא לא מרוצה ממהו שעשית לפני שבועיים, אבל הוא חיכה "להזדמנות המתאימה" שתשבו לדבר על זה. למה חמוץ? כי מהרגע הזה, אתה סוחב על הגב שלך את השבועיים האלה, שהוא הסתובב עם איזה ידע על משהו שהוא לא מרוצה ממנו, ואתה בכלל לא ידעת על זה. חוסר האונים הזה, שבכלל לא היתה לך אפשרות לתקן, או לדבר על זה, והידיעה שיש משהו שנסתר ממך, חמוצה ממש. חוץ מהחמוץ, זה גם הרבה יותר מסובך לתקן משהו שקרה לפני הרבה זמן. הזיכרון של רובנו לא עובד משהו. ולטס פייס איט, זה עושה שיהיה לנו פחות איכפת מזה. אז איזה פידבק עובד הכי טוב? כזה שהוא עכשיו ולא אחר כך.

וזה נכון תמיד - אבל על אחת כמה וכמה לאנשי פיתוח, שהעשייה שלהם נבנית וגדלה לאורך זמן, וקורית בשכבות. ואם יש שכבה אחת שלא עובדת כמו שצריך - זה לגמרי ישפיע על השכבות הבאות. מעבר לזה, אנשים טכנולוגיים מתמודדים עם פידבק שוטף כל הזמן. כל תהליך ה-Code Review, בנוי על היכולת להיות במעגל פידבק בלתי סופי של שיפורים וחשיבה משותפת על איך לעשות דברים טוב יותר. להיות מפתח או בחירה מודעת לחיות בעולם של שיפור מתמיד ופידבק שוטף.

בתור אחת המחלקות עם האימפקט המשמעותי ביותר בתוך החברה שלנו, הפוקוס על שיפור האיכות והיעילות של האינדיבידואל ושל הקבוצה כולה הוא תמיד. זה כשלעצמו הוא דבר חיובי, אך עלינו לשים לב שאנחנו גם ממוקדים בפידבק שאנחנו נותנים ולא סתם "זורקים" הערות. כמו בכל דבר, הרבה יותר פשוט "לשפר" מאשר לעשות מאפס.

האריזה היא חשובה, אבל אם יש משהו שיותר גרוע מלא לתת פידבק - זה לתת פידבק שמרוב שריככנו אותו, אי אפשר להבין מה המהות שלו. הרבה מנהלים חוששים להיות "קשים" מדי, ובמקום פידבק ענייני הולכים סחור וסחור עם אמירות כלליות. כמה פעמים קרה לכם, שישב מולכם מישהו שאתם מאוד אוהבים, או שמאוד ריחמתם עליו, או שלא רציתם להיכנס איתו לעימות? או שמרוב שכבר הרגשתם שנתתם פידבק, אתם לא רוצים להעמיס עליו, כי אתם מרגישים שזה יותר מדי? או שאולי חשבתם על זה יותר מדי, ולא בא לכם שהוא יילך לדבר עם החברים שלו על איזה בוס קשוח יש לו? אל תדאגו - אתם לא לבד. זה קורה לכולנו. ואז אנחנו מנסים לארוז, לעטוף, לצפות ולעדין את מה שיש לנו לומר, כדי שזה יירד קצת יותר חלק בגרון (או ככה לפחות נדמה לנו). האבסורד הוא, שזה הרבה פעמים עושה בדיוק את ההיפך.

הדבר הזה משאיר את שני הצדדים מתוסכלים - מצד אחד המנהל, שמה שרצה לומר דרש ממנו הרבה מאמץ (אין מה לעשות - לנסות ללכת מסביב למה שאנחנו רוצים, דורש עבודה). ומהצד השני, חבר הצוות - שלא לגמרי הבין מה רוצים ממנו, ולא מצליח להפיק לקחים מהפידבק שקיבל. וכך נוצר מצב שהצד השני מרגיש מתוסכל. יכול להיות שהוא רוצה לעשות את המקסימום כדי להשתפר, אבל למען האמת, זה דומה לנסות להכין עוגה בלי כוסות מדידה ומשקל - זה יוצא בערך. ואנשים אוהבים דיוק, על אחת כמה וכמה אנשים טכנולוגיים שחיים בעולם שבו יש דברים שעובדים ודברים שלא. כשאנחנו מנסים לא להיות "קשים", אנחנו הרבה פעמים עושים עוול גדול בהרבה לאנשים אותם אנחנו מנהלים.

לחוויה הקבועה יש גם ערך נוסף - כולנו מכירים סיפורים על חברי צוות שעובדים קשה, מנסים להשתפר, וכשמגיע הריוויזיה השנתית שלהם, הם פתאום שומעים בפעם הראשונה שהביצועים שלהם לא ברמה או בקצב הנדרשים. הם כמובן מאוד מופתעים ואומרים "אף אחד לא אמר כלום". ואז נכנסים למעגל בו המנהל אומר "אבל אמרתי". זוהי סיטואציה קשה לכל המעורבים ואנחנו רוצים לעשות כל מה שאפשר על מנת להימנע ממנה. זו גם הסיבה שבגללה החלטנו בטיילור לפני 3 שנים להחליף לחלוטין את תהליך ה-performance review feedback החצי שנתי בתהליך שבועי אמיתי. השורה התחתונה היא שאנשים אוהבים וודאות, ובכל שיחת פידבק שבועית שני הצדדים יודעים שיש להם מטרה משותפת לצאת מהשיחה כשהם בהלימה משותפת ויש להם תשובות ברורות על שתי שאלות: "מה מצופה ממני?" ו-"האם אני בכיוון?"

אחת המשימות הכי חשובות למנהלים טכנולוגיים היא לעבוד על האופן שבו הם מעבירים פידבק. זו משימה בלתי נגמרת, שתמיד אפשר להשתפר בה עוד ועוד (מצויין לאנשים שאוהבים שיפור מתמיד, לא?).

אז הנה ה"אני מאמין" שלנו לאיך לתת פידבק לאנשים טכנולוגיים:

1. ספציפיות

זה לא שיעור היסטוריה ולא שיעור פילוסופיה. פידבק טוב מדבר על נושא ספציפי שניגשים אליו, ואם יש לנו כמה נושאים - כדאי להפריד ביניהם ולטפל בהם בחתיכות קטנות וברורות ככל הניתן. למידה בשיעורים קטנים מייצרת אימפקט גדול. תחשבו על הפידבק האחרון שנתת למישהו. האם אפשר היה להפוך אותו ליותר ספציפי?

2. דוגמאות

לדוגמאות יש צד חיובי וצד שלילי. מהצד החיובי, הן עוזרות להבין טוב יותר על מה מדובר, ועוזרות לזכור יותר מאמירות כלליות. מהצד השני, הרבה פעמים דוגמאות מייצרות שיחה מאוד סובייקטיבית ויכולות להתפרש בהרבה דרכים. בשימוש לא נכון בדוגמאות, הן עלולות לגרום לצד השני להרגיש מותקף. שום מחשב לא יכול להחליף את שיקול הדעת של מנהל, וצריך להחליט באיזה דוגמאות להשתמש, ואיך להביא אותן לשיחה כדי שלא ייצרו מגננה אצל הצד השני.

3. זמנים קבועים

כשפידבק הוא קבוע ובאינטרוולים קצרים, כל הצדדים מתרגלים אליו. מרבית אנשי הפיתוח חווים פידבק על בסיס קבוע, כי בהרבה חברות יש תהליך מוסדר ל-code review. בעוד שהמופעים של code review מתורגלים ומתמקדים במה נעשה, המטרה צריכה להיות שונה, ולהתמקד באיך הדברים נעשים. אל תעשו שיחת פידבק רק כשמשהו קורה. תתרגלו את השיח הזה בזמנים קבועים ותהפכו את זה לשגרה אמיתית.

4. דו-כיוונית

לפעמים פידבק הוא חד-כיווני. משהו מדויק שצריך להעביר לצד השני בצורה הברורה ביותר. אבל רוב הפעמים, מדובר בשיחה דו-כיוונית. ומנהל טוב, נמצא כל הזמן במצב של האזנה כשהוא נותן פידבק, כי הוא מקבל מזה לא פחות ממה שהוא נותן. כשמגיעים לתת פידבק, עדיף לבוא פתוחים לשיחה וחשיבה משותפת, הרבה יותר מאשר עם הוראות ברורות.

5. שיקוף

הדרך הכי טובה להתקדם, היא לוודא ששני הצדדים מדברים באותה השפה. תמיד תשאלו את הצד השני שאלות שיקוף והבנה, כמו - איך אתה היית מנסח את מה שדיברנו עליו? מה הבנת מזה? מה אתה לוקח מזה? ומה היית עושה אחרת?

6. תחשבו קדימה

תמיד לקחת את הדברים לצעד הבא. לסיים שיחת פידבק תמיד במה שקורה הלאה. זה גם מייצר אופטימיות, כי יודעים מה הולכים לעשות, וגם מוטיבציה להתחיל לעשות את זה.

* חשוב - אל תחזיקו בצוות שלכם אנשים שלא יודעים לקבל פידבק. ותמיד תוודאו שכולם בצוות יודעים (ואם לא - למדו אותם איך) לתת פידבק, כדי שגם אתם תוכלו לגדול ולהתפתח ביחד איתם.

ועכשיו...תרגיל:

דמיינו את המנהל הכי טוב שהיה לכם. יש? מצויין.

תנסו לחשוב למה הוא היה המנהל הכי טוב שלכם? מה עשה אותו לכזה?

מבטיחים לכם, שכמעט תמיד, זה לא יהיה מישהו שהיה סופר נחמד וזהו. בדרך כלל, אלה יהיו המנהלים שדחפו, איתגרו, ונתנו פידבק משמעותי שגדלתם בזכותו. אלה המנהלים שהיו ממש משמעותיים עבורנו. וכשאנחנו חושבים על איזה מנהלים אנחנו רוצים להיות, זו צריכה להיות ההשראה שלנו. אבל בפועל, כשאנחנו נמצאים ביום-יום, אנחנו נופלים למלכודות הנחמדות, והרצון (המאוד טבעי) שיאהבו אותנו, וידברו עלינו דברים יפים עם חברים וקולגות כשאנחנו לא בסביבה. אבל זכרו - אם אתם אנשים סבבה, יאהבו אתכם בלי קשר לפידבק שתתנו. ואם תתנו פידבק מועיל ומפתח - אתם תהפכו להיות המנהלים האלה, שהם השראה, עבור האנשים שאיתם אתם עובדים. אין מה לעשות, בניהול מקצועי, יש בחירה להעדיף הערכה על פני חיבה. למרות שזה בכלל לא אומר שלא יחבבו אתכם, אבל לא כדאי שזה יהיה המנוע שלכם לפעולה.

9 מחשבות על ניהול אנשים טכנולוגיים

TLDR: מה סוד הקסם בניהול של אנשים טכנולוגיים ובמה זה שונה מכל ניהול אחר? בקיצור, 9 מחשבות על ניהול אנשים טכנולוגיים.

מקרה שבטוח לא קרה לכם: הייתם עם אוזניות על הראש, שלא ניגנו כלום, ורק רציתם שיחשבו שאתם שומעים משהו כדי שלא ידברו אתכם. נשמע מוכר?

אנשים טכנולוגיים עושים עבודה מאוד ייחודית שמשלבת בתוכה עבודה קשה, חשיבה ויצירתיות. ויש לה מרכיב רחב של כניסה לאיזור ריכוז. אינספור מחקרים מראים שיש משמעות לשעות שבהן אנשים יותר יצירתיים ומרוכזים. והדבר הזה מאוד אינדיבידואלי, אגב. יש אנשים שהשעות החזקות שלהם יהיו מוקדם בבוקר, ויש כאלה שהלילה זה זמן הפריחה שלהם.

בטיילור אנחנו לא מאמינים בשעות. אנחנו מאמינים בתפוקה. ודווקא בגלל זה, יש כמה דברים שצריך לעשות אחרת כשמגיעים לעבוד עם אנשי פיתוח:

1. ניהול היום

כן סטנד אפ? לא סטנד אפ? הרבה מנהלים מרגישים לא נוח עם ההתעדכנות, כי הם חוששים שזה יגרום לצוות להרגיש שעושים להם מיקרו-מנג'מנט. בעיקר אם רוצים לעשות סינכרון סוף יום, ובטח במודל היברידי כשאנשים עובדים מרחוק ולא רואים זה את זה. מהצד השני, הרבה מנהלים לא לגמרי יודעים (כשהם לא עושים את הסינכרונים) איך הדברים מתקדמים ואיפה יש תקיעות. ומשום כך לא יודעים מתי הם צריכים לצלול ולעזור. מהצד השלישי, (אנחנו בקוביית די-אנ-די), אנשים עובדים קשה, וכשאינן מישהו להתעדכן מולו זה יכול לייצר חוויה של בדידות "שלא רואים אותי". אלה מהפרספקטיבה הניהולית, אבל כמובן קיימת גם הפרספקטיבה הצוותית, כי גם ערך גדול לשיתוף בין חברי הצוות וכל כך הרבה פעמים קורה כשמפתח אומר: "וואלה, בדיוק נתקעתי עם זה לפני יומיים, בוא נשב אחרי זה ואני אעזור לך".

לכן, הנקודה החשובה פה היא איך הדבר הזה נעשה. דברו בקול רם על הסיבה שבגללה עושים סטנד אפ. שתפו גם פה, כמו בתהליכים אחרים - שהדברים הם דינמיים ופתוחים לשינויים ושתשמחו לקבל פידבק. בדקו עם הצוות אינדיבידואלית פעם בחודש-שלושה מה דעתם על הסטנד-אפ, מה עובד בו טוב ומה פחות. תבנו את העדכון בצורה שהיא יותר לפי כללים שהסכמתם עליהם מראש, ופחות על ידי שאילת שאלות ותשובות שמייצרת תחושה של חקירה. תקפידו לעשות את זה בזמנים קבועים ואל תבטלו. אם אתם באמת לא מנהלים שעושים מיקרו-מנג'מנט, השיחה הקבועה לא תרגיש כמו משהו שקורה רק כשמהו לא עובד.

*נקודה נוספת למחשבה (במיוחד בעידן של עבודה מרחוק) - לא בטוח שאתם צריכים עדכון סינכרוני (בלייב). עם מערכת ניהול משימות מסודרת ועדכונים שוטפים, למעט בעיות חריגות יכול להיות שתוכלו לוותר לחלוטין על שלב העדכונים היומי. כשהארגון גדל, שרשרת העדכונים היומית/דו-שבועית/ועוד נהיית בעייתית ומערבת עוד ועוד אנשים. עדכונים לפי צורך ובצורה א-סינכרונית מאפשרים גדילה משמעותית יותר ופחות זמן על "מטה-עבודה" (כלומר התעדכנות). דבר נוסף שעבודה כזאת מחזקת היא את תחושת האחריות האישית של חברי הצוות ו"חינוך" לעדכון בפוש ולא בפול. ז"א, ללמד את חברי הצוות לזהות באיזה מצבים עליהם לעדכן ולעשות זאת באופן יזום. אם נגיע לרמה גבוהה בגישה זו, נוכל לעבוד יותר עצמאית בתור מנהלים, ולתמוך בצוותים גדולים יותר. עם זאת, לא בכל ארגון זה עובד. ודורש משמעת נרחבת סביב שמירה על סדר וארגון בניהול המשימות האישיות, הצוותיות וכן הלאה. לכן ייתכן וזה יעבוד רק לחלק מהצוותים.

2. כמה שפחות פגישות

עוד לא פגשנו אנשי פיתוח שאהבתם הגדולה היא מרחב עבודה שעושים בו המון פגישות. ובכל זאת, יש פגישות שהיינו רוצים לעשות (כי יש בהן ערך). כל מנהל רוצה ליצור אווירה צוותית בפגישות צוותיות, לייצר למידה משותפת סביב case study מעניין, לעשות קפה משותף ולקשקש או להוביל רטרוספקטיב שלכולם יש מה ללמוד ממנו. תתייעצו עם הצוות - כמה זמן הם חושבים שצריך להשאיר לבלוק טוב של עבודה מבלי שזה יפגע בזרימה? מחקרים מראים שלוקח זמן להיכנס לdeep work. אנלוגיה מוצלחת ששמענו בעבר מדברת על כך שלהיכנס לעבודה משמעותית ויעילה זה כמו להירדם. עד שמצליחים, אם מישהו מעיר אותנו, אנחנו צריכים להתחיל מהתחלה. מאמר מוצלח אחר שעוזר להבין את ההבדל בין "יומן מנהל" ל"יומן מייקר" נכתב על ידי פול גראהם. בקצרה - מנהלים חיים בלו"ז של שעה בכל פעם. זה ממש העבודה שלהם להחליף קונטקסט כל שעה או אפילו חצי שעה. מייקרים לעומת זאת, חיים בקבועי זמן של יום או חצי יום. אחת השיטות הטובות להתמודד עם הפער הזה, הוא לרכז פגישות מוקדם בבוקר או לקראת סוף היום. בתור מנהלים, יש לקחת בחשבון שאם נפזר פגישות במהלך היום (נניח כי זה מה שעובד לנו בלו"ז) אנחנו צפויים לראות פגיעה משמעותית בתפוקה ובהנאה של הצוות. ומפתחים זקוקים לזמן של deep work. לכן כדאי לשים את הפגישות בקצוות של יום העבודה.

3. יצירת הזדמנויות לשיח משותף

בין אם אתם יושבים באופן ספייס יחד, ובין אם יש ערוצי סלאק שפתוחים לדיונים במהלך היום, תמיד קיים הדיסוננס בין להפריע למישהו כשהוא באמצע תהליך חשיבה מורכב, לבין היכולת להתייעץ, לשתף, ולהיעזר בחברי צוות אחרים. הכלל הראשון שצריך כדי לפתור את הדיסוננס הזה, הוא לדבר על זה בצוות. לכל צוות יש את הזרימה שלו. והיכולת לדבר על מה היינו רוצים, ומה עובד לנו ומה לא - בהחלט יכולה לעזור. אנשים מרגישים הרבה יותר נוח אם יישרנו קו לגבי שיחות במהלך היום. יש אנשים שמעדיפים לעזוב הכל ותמיד שמחים להיכנס לכל שיחה, ויש אנשים שהדבר הזה מכניס אותם לקושי. מצד אחד הם רוצים לעזור (ואנחנו יוצאים מנקודת הנחה שזה תמיד המצב אצל אנשים שגייסנו), אבל הקטיעה קשה להם. כשהנושא הזה לא מדובר, אחד מהשניים יקרה - או שהם יתפנו לעזור למרות שזה הפריע להם, וזה יעורר אצלם טינה שקטה כי הם יהיו מתוסכלים (מה שהרבה פעמים יכול לצאת החוצה אחר כך בהתנהגות פאסיב-אגרסיב), או שהם לא יעזרו, וירגישו רע עם זה שהם לא עוזרים כשצריך אותם. הפתרון לדבר הזה הוא בעיקר לעזור לכל אחד לנסח את מה שעובד לו. ולדעת להגיב בהתאם כשמבקשים ממישהו עזרה, והוא אומר - "לא עכשיו".

בהתאם לארגון ולסביבה שלנו אפשר להשתמש בשיטות שונות, למשל - להחליט שאם מישהו עם אוזניות, לא מפריעים לו באופן ישיר. או להחליט שתמיד שולחים קודם הודעה בסלאק, ואז מי שרוצה פוקוס יכול לכבות נוטיפיקציות. יש הרבה אפשרויות. מה שחשוב, זה לתקשר את ההחלטה, לחזור עליה, ולבדוק אם היא עובדת.

טיפ נוסף שנוגע באזור דומה - רעש בסביבת העבודה. דברו עם הצוות, וכל מי שחולק אתכם את אזור העבודה המשותף, והחליטו יחד על כמות הרעש המקובלת. זה יכול להיות אזור בלי מוזיקה, ללא אוזניות, או הגבלה של שיחות במרחב המשותף של עד 2-3 אנשים בלחש. או אזור חופשי שבו הכל הולך.

כל עוד מתקשרים את האופציות ובוחרים יחד, נמנעים מהרבה תסכול פוטנציאלי.

טוב...אנחנו מבינים שעד עכשיו כתבנו שצריך תקשורת, וזה לא נשמע פורץ דרך. אבל בוא נראה - האם דיברתם פעם על הדברים האלה בשיחה קבוצתית והחלטתם מה עובד לכם? ברוב המקרים, זה כזה פשוט, ודווקא כן פורץ דרך.

4. התמונה הגדולה

מכירים את המשל על העיוורים והפיל? המשל מגיע מהודו ומספר על קבוצה של עיוורים שכל אחד מהם ממשש חלק אחר של פיל, וביחד הם נדרשים להחליט איך הוא נראה. אחד מרגיש את השנהב ואומר: "פיל זה חד כמו חנית!" שני מרגיש את הרגל והברך ואומר: "פיל זה מחוספס ועבה כמו גזע עץ". שלישי מרגיש את הזנב ואומר: "פיל זה דק כמו חבל" ואחר מרגיש את החדק ואומר: "פיל זה עגלגל מתפתל כמו נחש". מפתח שרואה רק את מה שהוא עושה ולא מבין שיש פה פיל...בקיצור, הבנתם.

אנחנו מאמינים שמפתחים טובים, שעובדים על מוצר ביזנסי, צריכים להכיר את התמונה הגדולה. את היעדים, המטריות, המספרים. תשקיעו את הזמן בלתת קונטקסט לעשייה ולהבין מה המשמעות שלה על החברה כולה. עוד לא פגשנו מישהו שעשה עבודה פחות טובה כי הוא הבין למה הוא עושה את מה שהוא עושה. האחריות לייצר את ההבנה הזו נמצאת החל מהנהלת החברה, ועד לכל ראש צוות שצריך לתקשר משימות לחברי הצוות שלו. כשאנחנו מפנים זמן לתת את התמונה הגדולה, אנחנו מעבירים הרבה מסרים חשובים מתוך ה"אני מאמין" שלנו. אתם חלק אינטגרלי מהצוות שבונה פה חברה, אתם חשובים למשימה, אתם לא עובדים בוואקום, המשימות שלכם בונות עולם תוכן שמקדם את הסיבה שבגללה הקמנו את החברה הזו. אי אפשר לדבר בסיסמאות על הצורך של אנשים באימפקט, בלי לתת להם את התמונה הגדולה והמלאה. בנוסף, כשאנחנו נותנים לאנשים את התמונה המלאה, הם לוקחים אחריות על ההשפעה של המשימה שלהם ולא רק על הדלוור שלהם. בנוסף, כתוצאה מכך, נוצרות מערכות יחסים טובות יותר עם הפרודקט, דיזיין, סילס ושאר המחלקות.

5. הימנעו ממיקרו-מנג'מנט

עוד לא פגשנו מישהו שאמר לנו שהוא אוהב לנהל בשיטה הזו, אבל וואלה, יש לא מעט מנהלים שכן עובדים ככה. אז מה הקטע? הרבה פעמים, מיקרו-מנג'מנט נובע מחוסר ביטחון, או חוסר ניסיון של המנהל, שמתקשה להוביל סיטואציה מסוימת אל עבר היעד ומפחד לשחרר. יש הרבה דרכים להימנע ממיקרו-מנג'מנט, אבל לפני שניגע בהן, חשוב שנבין למה זה עושה כל כך רע (בעיקר למפתחים). שני ערכי ליבה שאנחנו רוצים שהמפתחים שלנו יחזיקו הם עצמאות ואמון. מיקרו-מנג'מנט פוגע בשניהם. זו שיטה שממש מניחה בסיס לייצר את ההפך ממה שאנחנו מאמינים בו. כדי לנהל מפתחים לאפקטיביות, צריך לתת להם את המה ולא דווקא את האיך.

כמה טיפים שיעזרו להימנע ממיקרו מנג'מנט:

- + נסו לוודא שה"מה" ברור ככל הניתן. ראשי צוותים עובדים לא מעט עם הפרודקט, וככל שהמשימה שלהם תהיה יותר ברורה, ככה גם העבודה העצמאית שלהם תהיה טובה יותר.
- + תנו לאנשים שלכם להגדיר את הדד ליין איפה שאפשר. תעזרו, תאגרו - אבל שימו אותם אחראים.
- + נסו לשים את נקודות הממשק להתעדכנות בזמנים קבועים. זה מוריד את התחושה של גחמה של מנהל שרוצה עכשיו לדעת מה קורה, כי פתאום הוא נלחץ, או כי שאלו אותו משהו והוא לא ידע לענות. זמנים קבועים קורים מעבר לכל סיטואציה. כל עוד אתם בתחילת הדרך המשותפת שלכם יחד, נסו שהאינטרוולים בין זמני ההתעדכנות הקבועים יהיו קצרים, ועם הזמן (והניסיון) תרגישו נוח לרווח אותם.
- + השתדלו לאתר צמתים שבהם אתם יכולים להבין אם נדרש שינוי בגישה. תיצרו שיחות המסבירות למה אתם מבקשים לשנות עכשיו את האופן שבו משהו נעשה. נסו להשאיר את הצמתים האלה במקומות שמשאירים באפר לטעויות. ככל שיהיה פחות לחץ על המשימה, ככה יהיה לכם יותר מרווח לנהל שיחות ולא "לירות הוראות".
- + כל סיטואציה היא הזדמנות טובה ללמידה - תשקיעו את המאמץ ללמד כמה שאפשר את הצד השני מתי להרים דגלים ואיך.

6. חוזקות

כמה ספרים נכתבו על עבודה דרך חוזקות? מלאן. כמה מאיתנו מקבלים פידבק, וקצרים מדי כדי להקשיב לחלק הטוב, כי אנחנו רוצים כבר לשמוע מה לא טוב? מלאן. אנחנו גדלים בסביבה הבנויה להתעסק במה שלא עובד, ונועדה לעבוד על החולשות שלנו במקום לשפר את החוזקות שלנו. כולנו חונכנו במערכת שתומכת בשיפור החולשות שלנו ומתעסקת בהן בלי סוף. לכמה מכם היה ביסודי מורה פרטי למקצוע שאתם מעולים בו? לכמה מכם היה מורה פרטי למקצוע שהתקשיתם בו? אנחנו משקיעים המון אנרגיה במה שאנחנו מתקשים בו. דברו עם האנשים שלכם על החוזקות והחולשות שלהם בפתיחות. תעצרו מדי פעם לשמוע מהם על ההנאות והמוטיבציות שלהם: ממה הם נהנו במשימה האחרונה? מה גרם להם להתרגש כשהם באו לעבודה? איפה הם הרגישו שהם הביאו את "האני הכי טוב שלהם"? מאיזה משימות הם היו מעדיפים להיפטור? מה הם מרגישים שתוקע אותם וממה הם פחות נהנים? נסו למצוא איתם את התמהיל הנכון. 80% מהזמן נסו להקצות משימות עבודה דרך החוזקות והמוטיבציות שלהם, אבל כל הזמן לדחוף לעשות גם קצת ממה שלא. שלא יתנוונו (פה ה-20% הנותרים). מפתח טוב חייב להתמודד עם החלקים שפחות חזקים אצלו. גם כדי שלא יהפכו להיות חסם של משימות שהוא לא יכול לקחת, וגם כדי שהחשש לקחת משימות שהוא פחות טוב בהן, לא יהפוך להיות כל כך גדול מרוב שהוא לא מתרגל אותן וירתיע אותו ממש.

7. דוגרי - בישירות ובכנות

זה נכון לכל חבר צוות, ועל אחת כמה וכמה למפתחים. השפה שמפתחים מדברים בה ביום יום היא ברורה. אמצו לעצמכם סגנון שיחה שדומה לעולם התוכן היומיומי של מפתחים - ברור, מדויק, ענייני. נסו ללמוד כיצד לשוחח בצורה עניינית על הדברים, ולמדו את חברי הצוות שלכם להגיב אליכם באותו המטבע (ואם אתם רוצים לצלול קצת יותר לעומק, יש עוד תוכן מורחב על הנושא הזה בפרק של "פידבק").

8. פירגון

שתפו החוצה את העשייה של האנשים שלכם. לרוב, העשייה של המפתחים לא נמצאת בשורה הראשונה של מחיאות הכפיים הארגוניות מכל מיני סיבות. קודם כל, פחות קל לשאר הארגון להבין מה הם עושים, ולכן גם יותר קשה להריע. הסיבה השניה היא שיש אנשים אחרים שהעשייה שלהם הרבה יותר בולטת בתוך הארגון, ויותר רועשת. והסיבה השלישית, היא שהרבה פעמים זו עבודה שנעשית במשותף, ולא תמיד יש שם אחד להלל ולהרים. לכן, מנהל טוב הוא כזה שיודע להיות שופר לשאר הארגון, ולעזור לסביבה לראות את כל הטוב הזה ולהחצין אותו במפת ההוקרה הארגונית. אין מישהו שלא צריך שיראו אותו, יעריכו אותו, ויקירו את העשייה שלו. מפתחים הרבה פעמים צריכים שהמנהלים שלהם ישמשו עבורם מגבר לעשייה. זה עליכם.

9. אל תיתקעו

פיתוח הוא תפקיד לומד, נושם, שגדל באופן מתמיד. תייצרו לכם את המהלכים שבהם אתם ממשיכים יחד עם הצוות ללמוד להעשיר זה את זה. אל תתפשרו על סטגנציה. אחד החסמים הגדולים, הוא שאנחנו מתפתים למהלכים גדולים: כנס גדול, אירוע למידה משמעותי וכיוצא בזה. לעומת זאת, מנהל שיודע לשתף את הצוות בפודקאסט מעניין שהוא שמע, ולנהל עליו שיחה ערה בישיבת צוות - עושה קסמים. ומנהל שיודע לגרום לצוות שלו לעשות את אותו הדבר ולקחת חלק בהתפתחות המשותפת הזו - הוא כבר קוסם על. זה קל, זה זמין, וזה מיידי. וזה בעיקר משהו שתלוי בכם, הרבה לפני משאבים, תקציבים ואירועים גדולים.

באופן כללי - שאלו את עצמכם תמיד, למה אתם עושים/מבקשים את מה שזה לא יהיה? למי זה תורם? מה זה בא לפתור? מה יקרה אם לא נעשה את זה? ובזמן שאתם עונים על השאלות הללו, התמקדו בדברים שמקדמים את העשייה והיעילות של חברי הצוות שלכם. השתדלו לבחור בפעולות שמטפלות - עד כמה שאפשר - בצורה ישירה בדברים שלשמם החלטתם לעשות זאת.

ג'וניורים: כן או לא? (בקצרה...כן!)

TLDR: ג'וניורים - מתי הם טובים לנו? איך מכניסים אותם לעניינים? ואיך מגדלים אותם בתוך החברה? כמה עצות על תהליכי העבודה, וגם - מתי יודעים אם זה לא מתאים?

את מה אנחנו זוכרים יותר - את ההימורים שלקחנו והצליחו לנו? או את ההימורים שלקחנו ולא הצליחו לנו? בדרך כלל, התשובה תהיה ההימורים שהפסדנו בהם. בני אדם שונאים להפסיד מטבעם. וכשזה מגיע להימורים, אנחנו מעדיפים למזער סיכויי הפסדים מאשר למקסם סיכויי הצלחה. כשמגייסים ג'וניור, לוקחים הימור. מאמינים בפוטנציאל וביכולות, ומגייסים אותו על בסיס הבטחה שאמורה להתממש בשלב מאוחר יותר.

לגייס ג'וניורים לארגון נותן ערך בהמון מובנים: זה נכון לארגון לווסת את הידע שנכנס, יש ערך רב במישהו שלומד איך לעשות את הדברים בדרך שלכם מהתחלה, זה מאפשר להגדיל את מצבת כוח האדם יותר בקלות, וגם ברמה הבסיסית ביותר - אם לא נגייס את הג'וניורים ונכשיר אותם - איך נגדיל את קהילת המפתחים שלנו?

כשאתה מסתכל על סל המשאבים שיש לך בצוות, שווה להשקיע חלק ממנו בהימור על משהו שאתה מאמין ימקסם את עצמו לאורך זמן. ואל תשכח - יש לך את כל המשאבים כדי לגרום להימור הזה להצליח.

לאורך הדרך למדנו הרבה על פיתוח ג'וניורים. ודבר אחד חשוב במיוחד ששווה להתייחס אליו לפני שאנחנו צוללים לאוסף של עצות מעשיות: לג'וניורים תמיד צריך להיות לא נוח. לא מילולית כמובן, אלא הם כל הזמן צריכים לנסות ללכת בנעל שגדולה עליהם במידה אחת. משמע, בפיתוח ג'וניורים - לא טוב שיהיה איזור נוחות. בכל פעם שאנחנו מוצאים את עצמנו מגיעים לזה - זה סימן שצריך להגביר את הקצב.

אז..איך מפתחים ג'וניורים?

באופן כללי, ה"אני מאמין" שלנו, הוא שמנהלים הם מורים ומדריכים יותר מאשר מפקדים. זה אומר שלא צריך לומר לאנשים מה לעשות, כמו שצריך לגרום להם להבין למה הם צריכים לעשות את הצעד הבא. זה לוקח יותר זמן, וזה לא תמיד קל, אבל בעינינו, יש בזה הרבה יותר ערך "מלירות" הוראות, ולעקוב אחרי ביצוע.

הרבה פעמים אנשים חושבים שכשרוצים לפתח ג'וניורים, בגלל הניסיון המועט שלהם, הם צריכים סגנון ניהול שונה, שהוא יותר הוראתי ופחות הדרכת. אנחנו מאמינים שעם כל מפתח בארגון - הגישה הזו צריכה להיות זהה. ומוטב להתחיל ללמוד את ההתנהלות מהיום הראשון, מאשר לחכות לרמת בשלות מסויימת, ואז להתאים אליה ניהול אחר. ישר כשמתחילים לעבוד יחד לומדים מי נמצא מולנו, ומתחילים לצבור יחד דרך, כדי להבין אם אנחנו בכיוון הנכון או לא.

למדו אותם ללמוד

ג'וניורים מגיעים, על בסיס הגדרתם, עם מעט ניסיון. וניסיון עושה הרבה דברים יפים. הוא מקצר זמנים לפיתרון בעיה. והוא גם עוזר לייצר פתרונות טובים יותר. אז מה קריטי כדי שג'וניור יצליח? יש הרבה דברים חשובים: ידע, התנסויות, פידבק. אבל על כל אלה, יש איזו שמיכה מלמעלה שקשורה להכל - ביטחון. ביטחון בעצמו ותחושת מסוגלות יעזרו לו ללמוד, לחקור, לעשות, וגם לטעות.

מנהלים צעירים הרבה פעמים מוצאים את עצמם מתלבטים: "איך אני יכול להקנות ביטחון לחבר צוות ג'וניור שהגיע?" בניגוד למה שעובד במחלקות אחרות, ריבוי שבחים ועידוד פחות יעזרו כאן. הדרך הכי טובה לייצר אצל ג'וניור ביטחון היא **ללמד אותו איך ללמוד**. עובר לך בראש המשפט הזה שמדבר על זה שצריך לתת חכה ולא דגים? אז כזה. רק שאנחנו מאמינים שצריך ללמד אותם לבנות את החכה.

"ללמד איך ללמוד" יש 3 שלבים:

- + ביסוס מוטיבציה
- + מיפוי ערוצי למידה וסולם אסקלציות
- + תחקיר

בואו נפרק שלב שלב:

מוטיבציה

המנוע פה מוזן מהחשיבות. כדאי קודם כל ולפני הכל, ממש בתחילת הדרך המשותפת יחד (ואפשר גם להתייחס לזה עוד בתהליכי הראיונות) להבהיר את החשיבות של היכולת ללמוד, ועד כמה היא תהיה בעלת ערך לקבוע הצלחה.

מיפוי

למפות יחד את כל ערוצי הלמידה השונים שעומדים בפנינו: למשל, גוגל (אובייסלי) ולשאל חברי צוות אחרים (אובייסלי 2), אבל גם מקורות פנימיים בארגון, **learning centers** וכו'.

אחרי שמיפיתם את הערוצים, חשוב לייצר **סולם אסקלציות**. כי בלמידה, הדרך הכי מהירה היא לשאול מישהו מנוסה, אבל כמו שאנחנו יודעים, זה בעל ערך הרבה יותר קטן מאשר להצליח ולמצוא את התשובה בעצמך. (וזה נכון גם בהורות, ביי דה ווי).

אחרי מיפוי סולם האסקלציות, נכנסת פה חתיכה של הקצאת משימות -
לפחות 2-3 שבוע - שהן למידה מלאה, וכוללות:

תחקיר

הנה כמה שאלות לדוגמא, עבור שיחה על משימה שקיבל חבר צוות ולא ידע איך לעשות:
מה היו הדברים במשימה שלא ידעת מספיק טוב לפני שניגשת למשימה?
איפה בחרת להתחיל להשלים את הפער? ולמה?
מה למדת בערוץ הראשון?
האם השתמשת לאחר מכן בעוד ערוצים? האם הם תרמו לך במשהו? מה למדת בהם שלא
למדת בערוץ הראשון?
איך אתה מתעד את הידע שצברת עכשיו, כך כשתצטרך אותו שוב הוא יהיה קל לשליפה?
אם יש חשק, כדי לנסות לגרום לחבר הצוות להסתכל מבחוץ על תהליך הלמידה שלו עצמו, על
ידי זה שהוא משליך אותו על מישהו אחר: מה בתהליך הלמידה עבד לך טוב ואתה חושב שאפשר
יהיה להעביר לחבר צוות הבא שייתקל במשימה כזו בלי ידע מוקדם?

מקצוענות מהרגע הראשון

הרבה פעמים נדמה לנו שג'וניורים צריכים סטנדרטים אחרים, כי אנחנו לא רוצים "להפיל עליהם
יותר מדי על ההתחלה" או כי נדמה לנו שזה לא מאוד משנה, כי יש דברים שאפשר להשלים אחר
כך. אבל האמת היא שהרגלים ותרבות עבודה מתחילים להתהוות כבר מהיום הראשון - בין אם
נרצה ובין אם לא. וכשאנחנו לא שמים דגש על משהו - אנחנו בעצם מלמדים שהוא לא מספיק
חשוב. כשאנחנו אומרים על משהו שנלמד אותו אחר כך - אנחנו אומרים שזה לא מצופה ממך
עכשיו, וזה בסדר שלא לכולם יהיה את זה. אנחנו מנחילים הרגלים ותרבות לא רק על ידי מה
שאנחנו מתעקשים עליו, אלא גם על ידי מה שאנחנו לא מתעקשים עליו.

אז מה אנחנו חושבים שנכון לבסס כבר מההתחלה ואיך אפשר לעשות את זה?

איכות

הכוונה באיכות אצלנו, היא האיכות הנדרשת. איכות תמיד נגזרת ממה שאנחנו מנסים להשיג.
וחשוב לנו ללמד שתי גישות מחשבתיים: האחת, נוגעת למחשבה סביב מה אנחנו עושים ולמה,
והשניה נוגעת לעיקרון של גאווה במה שבנית. בעולם האמיתי, שום דבר לא מושלם, לא אנחנו,
לא הקוד שאנחנו בונים, לא המוצר שלנו ולא הארגון. החיים אפורים. אז איך נדע לאיזו איכות
לכוון? יש פה איזה פארטו להשקיע אנרגיות כדי לשפר את החוויה שתהיה יותר חלקה, ניטור יותר
טוב, יותר טסטים, תמיכה ביותר סוגי מכשירים או רזולוציות. איכות לאו דווקא אומרת פחות
באגים. אבל צריך לבחון במה צריך להשקיע כדי לקבל רווח גדול. אבל איך נדע מה האיכות
הנכונה למה שאני בונה? הרמה הנדרשת תלויה במה שאנחנו מנסים להשיג. וזו חשיבה שצריך
להתחיל וללמד אותה כבר מההתחלה.

לדוגמא, אם המוצר הוא פנימי לארגון, ואפשר להריץ אותו רק על מחשבים של הארגון, כנראה שאין סיבה לתמוך בו ברזולוציה של מכשירים סלולריים. אם אנחנו מוציאים שירות שמיועד רק לארה"ב, אנחנו יכולים לבחון מה רוחב הפס הנפוץ, ולהתאים את האפליקציה שלנו לשם. במדינות עם חיבורים יותר חלשים - צריך להתייחס גם לזה.

אז בואו נדבר על באגים. כאן צורת המחשבה שנרצה ללמד, היא שיש אלמנט של גאווה עצמית. אם היינו שף, לא היינו מכינים תבשיל ומגישים אותו מבלי לטעום אותו ולראות שיצא מצויין. אם אתה מייצר מוצר - תבדוק אותו בעצמך כמו שף. כל דבר שאתה יכול למצוא בעצמך בזמן סביר - יימצא גם על ידי מישהו אחר. אבל חשוב שאתה תתפוס לפני את הדברים לבד. בהתאם לזה הדרישה של האיכות. אנשים עם ניסיון מועט מקבלים משימה ומבצעים. כשאתה מבין את ה"למה", אתה יכול להבין גם מה האיכות שנדרשת.

עמידה בזמנים

חייבים למצוא את המקום שבו יש אחריות מלאה על התוצר אצל מי שכותב קוד ולתת לו לעמוד בזמן. התגובה שלכם בפעם הראשונה שזה קורה מגדירה מה מצופה הלאה. אנחנו ממש לא אומרים להתחרפן מהאיחור הראשון, אבל אנחנו כן אומרים לא לתת לזה לעבור על סדר היום. זה ממש לא אומר שאנחנו צריכים לכעוס או להתרגז. אבל אם נשקיע זמן לדבר על זה - נעביר את המסר שזה חשוב. בתוך כל הלחץ, הרבה פעמים מנהלים, שרואים איחור בדילבור, עוברים על זה מאוד מהר כדי לקדם את הפתרון. הפועל היוצא של זה, הוא שזה לא נורא. וזה לא מסר שנכון להעביר אם אנחנו רוצים שעמידה בזמנים תהיה בעלת ערך בארגון.

אחריות

אחד הדברים שאנחנו הכי רוצים לבסס אצל מפתחים, הוא את העצמאות שלהם - גם בדברים הקטנים. עצמאות בהגדרה שלה מגיעה עם אחריות. גם אם היא לא רחבה בהתחלה, וגם אם אי אפשר לתת אחריות על משימה ברמת הפיצ'ר. בכל משימה, כדאי למצוא מידה מסוימת של אחריות. אחריות יכולה להיות, ובטח בתחילת הדרך בארגון, דבר חמקמק. הרבה פעמים מגלים אותה רק כשמשהו לא עובד, וצריך לראות איפה הייתה הבעיה ואיך מוודאים שלא תחזור. במצב כזה, אנחנו מגלים את האחריות תחת מטריה של משהו שלא עבד. וזה מבאס. אנחנו מאמינים שאחריות צריכה להיות מדוברת ומתוקשרת. כשאנחנו נותנים אחריות - לקרוא לילדה בשמה. מה האחריות, מה היא כוללת ומה מצופה בהקשר שלה. אנחנו גם מבקשים לעדכן כשסיימנו משימה - כדי שנוכל להעריך את ההצלחות של משימה שהייתה תחת האחריות של מישהו ונעשתה בהצלחה (שזה לרוב מרבית המקרים, והם מסמנים התקדמות). גם כשאנחנו אחראים על משהו, אנחנו אחראים לוודא שהוא קרה מ-א' עד ת'. ואנחנו אחראים לעדכן שהוא הסתיים ואפשר לעבור לדבר הבא.

תקשורת

אנחנו לגמרי מתנגדים לסיפורים על זה שאנשים טכנולוגיים הם לא אנשים תקשורתיים. פסו מן העולם הימים שסיפרו שמפתחים יושבים באיזה חדר שקט בקצה המשרד ואסור לדבר איתם, כי הם טיפוסים גאונים כאלה, שהחבר הכי טוב שלהם הוא המחשב. זה אולי עדיין קיים בכל מיני מקומות, אבל בטח לא אצלנו. זה לא ה"אני מאמין" שלנו. מפתחים נמצאים בליבת העשייה, וככאלה, הם צריכים להיות אנשים עם יכולות תקשורת מאוד גבוהות. הדבר המשמח לגבי תקשורת, הוא שאם מישהו הגיע מהבית עם בסיס טוב, את רוב הדברים שהיינו רוצים לראות אצל מפתח אפשר ללמד. וזה חלק משמעותי בפיתוח של ג'וניורים בתחילת הדרך (ובואו נאמר את האמת - זו עבודה שאינה פוסקת, לאורך כל הדרך. גם עם סניורים, מנהלים ומנהלים בכירים).

אז על מה בכל זאת כדאי לשים דגש אצל ג'וניורים בעולמות התקשורת?

איך לבקש עזרה?

לכל ארגון יש את המומנטום שלו. יש ארגונים שבהם מקובל בכל רגע נתון לעצור מישהו בזמן שהוא עושה משהו ולשאול אותו שאלה, יש ארגונים שבהם נהוג לשלוח הודעה בסלאק ולחכות שהצד השני יתפנה לעזור. ויש ארגונים שיש שעות ייעודיות לדבר הזה. **לא משנה מה ההרגלים של הארגון שלכם - צריך לדבר עליהם בקול רם ולהגדיר אותם.** הדבר האחרון שהיינו רוצים, זה שמישהו יעמוד מול בעיה שהוא לא מצליח לפתור, ופשוט ישב ויחכה כי הוא לא יודע מה הכללים התרבותיים של איך לגשת ולבקש עזרה. אבל פה לא הכל נמצא בצד של מי שזקוק לעזרה. הדרך הכי טובה ללמד לבקש עזרה - היא דוגמא אישית. מפתח שרואה את ראש הצוות שלו מבקש עזרה (או מספר לו מה הוא עשה כשהוא נתקע בבעיה עם הבילד בשבוע שעבר, ואיך הוא פנה לבקש עזרה) לומד מזה לגיטימציה ופרקטיקה. נושא נוסף חשוב בעולם הזה הוא ללמד מה השלבים הראשונים, טרם ההגעה לבקשת עזרה. מה הם השלבים שלא דורשים משאבים מאך אחד, ואפשר לבדוק אותם לפני שמגיעים לבקש עזרה (למשל יוטיוב, גוגל, גיטהאב).

מתי להרים דגלים בזמן?

אין אחד שלא יודע שזה אחד הנושאים החשובים. הרמת דגל בזמן תעזור לנו לתקן תוך כדי, במקום בדיעבד. מושלם. הבעיה היא שלהרמת דגל יש כמה אלמנטים. הרמת דגל כשלעצמה לא מאוד מעניינת, אלא אם היא נעשית בזמן. בגלל זה תמיד אומרים "להרים דגלים בזמן" ולא רק "להרים דגלים". קודם כל, צריך להתחיל בלדבר על המחויבות להרים דגל. והמחויבות הזו נמצאת אצל מי שמחזיק באחריות. בדרך כלל, כדי ללמוד את הנושא הזה, אנחנו רואים שני סוגים של אנשים:

הסוג הראשון - כאלה עם חוסר ביטחון גדול, בדרך כלל בתחילת הדרך, שמרוב לחץ מרימים את הדגל קילומטרים לפני המטרה. הם חוששים שלא יצליחו לעמוד במשימה, חוששים לטעות, ומעדיפים להתריע מראש. הבעיה עם ההתראה, שהיא כל כך רחוקה מהיעד, שהיא הופכת להיות לא רלוונטית.

הסוג השני - הנחויים, שלא מרימים את הדגל בזמן, כי עד הרגע האחרון הם ממש בטוחים שיש להם עוד מה לעשות כדי להשפיע על המצב ולשנות אותו. אז יוצא שמתוך המון כוונות טובות, עד שהדגל הורם, הם פספסו את ההזדמנות לשנות משהו בזמן.

כך או כך, המקרים האלה יקרו. אין מצב שלא. אבל היופי במנהל שמלמד תקשורת טובה, הוא להשתמש בסיטואציות האלה כדי ללמוד מהן יחד עם הצוות שלו. במהלך הלמידה, כדאי להתייחס לסיבה שבגללה הדגל לא הורם בזמן, ולנסות להבין יחד איך אפשר למנוע את זה לפעם הבאה. בקיצור - לדבר את זה, שוב ושוב - ושוב.

לתקשר את מה שאתה עושה:

מרבית האנשים יגידו שהם לא אוהבים לעבוד בשיווק עצמי, שהם מעדיפים לעשות את העבודה ולהצליח בה וזה מספיק. העניין הוא (ועל אחת כמה וכמה במודלים היברידיים, או בחברות כמו טיילור, שמלכתחילה עובדים בגישה של **Work from Anywhere**) שזה יכול להתפספס. ואחד הדברים הקריטיים בנוגע לג'וניורים, הוא להבין את קצב ההתקדמות והעשייה מהר מאוד. כמו שאמרנו, כשמגייסים ג'וניור, מגייסים אותו מתוך אמונה בפוטנציאל שלו וביכולות שלו. גייסנו אותו על בסיס הבטחה. אחרי זה, יש לנו פרק זמן של סבלנות ואורך רוח, לראות אם ההבטחה הזו מתממשת. בדרך כלל, פרק הזמן הזה גם מוגבל בזמן, וכמנהלים, אתם תצטרכו לדעת מה הקצב. וככל שיש מישהו שיש לו יותר מה ללמוד, אתם רוצים לדעת במה הוא משקיע את הזמן שלו כדי להבין את התמונה.

כדאי להימנע מאמירות כלליות של "תתקשר לי מה אתה עושה", או "תגיד לי כשסיימת את המשימה", אלא ממש להנחות באיזה מוקדים הייתם שמחים לקבל מידע בדחיפה מהג'וניור שאתם מפתחים. זה יכול להיות מבוסס על נקודות בזמן (וכל עוד הן קבועות זה הופך את זה ליחסית קל), וזה יכול להיות מבוסס על נקודות של התקדמות (וגם הן צריכות להיות כמה שיותר מדויקות ולא אמורפיות בהגדרה שלהן).

איך לא להרוג מוטיבציה?

ג'וניורים מגיעים עם המון מוטיבציה ללמוד, להתפתח, לעשות יותר. להתעסק בדברים שמעניינים ומרגשים אותם, לקבל משימות בעלות השפעה. לפעמים, מתוך המוטיבציה, אנחנו רואים שצריך קצת להרגיע. בין אם זה בקצב, או במורכבות של המשימות, בגודל שלהן, או בדברים שבהם הם מתעסקים. זה עניין מסובך, כי יש קו דק בין לנווט מוטיבציה לבין להרוג אותה. כשאנחנו נו מאמינים שג'וניור עוד לא שם, והוא צריך עוד תרגול וניסיון להתאמן על משימות שהוא חושב שמשעממות, בזמן שהוא מאמין שהוא כבר "אחריהן", ויכול לעשות דברים אחרים, אנחנו רוצים לוודא שאי ההסכמה הזו, לא הורגת את המוטיבציה שלו. הניסיון מלמד אותנו שתקשורת מאוד כנה וברורה לגבי העניין הזה עוזרת מאוד, יחד עם שרטוט מסלול מאוד ברור. אם לא עכשיו - מתי כן? ה"כן" יכול להיות בעוד X זמן שעובר והוא יכול להיות בעוד X משימות.

אנחנו גם ממליצים להיות מאוד בבחינה עצמית, ולראות שגם אנחנו, כמנהלים, לא מפחדים להתקדם ולתת עוד משימות. מאוד יכול להיות שאם מי שמולנו אומר שהוא יכול יותר - הוא אכן יכול. אנחנו יודעים הרבה, אבל אנחנו ממש לא יודעים את הכל, וחשוב להיות בקשב למי שמולנו, ולראות אם מתי שהוא אומר שהוא יכול ורוצה יותר - זה אכן המצב. מתי זה קורה בגלל אגו ורצון להתקדם כדי "להיות כבר שם", ומתי הוא באמת כבר הבין את המקום שלו וצבר מספיק כדי לעלות עוד שלב. איפה שאנחנו חוששים לקחת סיכונים, תמיד אנחנו יכולים לנסות ולדאוג שתהיה לנו סביבה מבוקרת כדי לצמצם את המחיר אם משהו משתבש.

ציפיות ברורות:

מנהל צריך לעשות תיאום ציפיות בשני מישורים: מול חבר הצוות הטרי (AKA הג'וניור), ומול חברי הצוות שלו.

מול מי שאנחנו מפתחים חשוב להגדיר שני סוגים של ציפיות:

+ ציפיות קצרות מועד - שמסתכלות שבוע קדימה.

+ ציפיות רחבות - שמסתכלות חודש או סוף פרויקט קדימה.

כל דבר מעבר לזה, בתחילת הדרך, הוא פחות משמעותי, כי אנחנו עוד לומדים אחד את השני. הדבר הזה נכון לתקופת ההסתגלות של בין רבעון לשני רבעונים ראשונים. אחרי זה אפשר להרחיב. הרבה פעמים אנחנו רואים שמנהלים חושבים שפיתוח של ג'וניורים והגדרת ציפיות תעשה באופן שהוא יותר מדי כללי וקשה לתרגום של היום יום. כזו שמסתכל רחוק מדי (רבעון או חצי שנה קדימה). בסיטואציה כזו, אם הגדרנו ציפיות רחוק מאוד, קשה לאנשים לעשות reverse engineering למה שעושים מחר או בשבוע הבא כדי לעמוד בתמונה הגדולה, וזה מבלבל. זה גם יכול ליצור מצב שבו אנחנו מגלים שעבר רבעון, ויש חוסר הלימה בין איך שהמנהל רואה את הדברים ואת ההתקדמות, ואיך שחבר הצוות רואה אותם. במצב כזה, אנחנו פתאום מרגישים שהתבזבז לנו רבעון, כי אנחנו מגלים על זה מאוחר מדי. כי את חוסר ההלימה הזה אפשר היה לתקן כבר ממש.

ומה לגבי שאר הצוות? הרבה פעמים אנחנו חוששים להביא ג'וניורים לצוות בגלל המעמסה שהדבר הזה יש על הכתפיים של המפתחים המנוסים שלנו. לכן מאוד חשוב להגדיר מראש ולברר כמה אנחנו מוכנים לספוג, ואיך זה צריך להיראות. תיאום ציפיות מול חברי הצוות צריך לכלול שיחה מאוד כנה על איך כל אחד בקבוצה יכול לעזור לג'וניור שלנו להצליח. כמה זמן אנחנו מצפים מהם להשקיע במי שפחות מנוסה? באיזה אופן היינו רוצים לראות אותם עוזרים לו? מתי אנחנו רוצים שירימו לנו דגלים? (אם למשל, ההתקדמות מאוד איטית, או שיש פער בתפיסת עולם, או שחלילה יש מצבים של הסתרה כשהוא מפשל או לא יודע משהו). ומתי אנחנו חושבים שזה לגמרי בסדר לומר להם שיסתדרו לבד, או לשלוח אותם לעשות עוד עבודה, עוד מחקר או עוד למידה לפני שעוזרים להם לפתור את הבעיה. כי מעבר לזה שכולם רוצים להיות מאוד נחמדים - יש פה אינטרס משותף להגדיל את מרחב העצמאות.

השיחה הכנה והפתוחה צריכה לא רק לתת את ההגדרות שלנו כמנהלים, אלא לשמוע גם משאר חברי הצוות איך הם מרגישים שהם יכולים לתרום לחבר צוות החדש. אנחנו רוצים לייצר שותפות ביחס להצלחה של כל חבר צוות חדש בארגון, ולוודא ששאר חברי הצוות רוצים לקחת חלק בפרויקט ובהצלחה שלו. אם אנחנו מבססים את העובדה שזה חלק מהתפקיד של כל אחד, כמפתח מנוסה בארגון, שזה לא מובן מאליו ושזה חשוב ובעל הערכה, אנחנו מגדילים את הסיכוי להצלחה.

פנינה אחרונה, וחשובה לא פחות (אם לא יותר):

להתחיל בארגון חדש, ועל אחת כמה וכמה כשאתה יודע פחות מאחרים, זה מקום מאוד בודד. צריך להכיר בזה, ולשים לב שמי שנכנס לארגון - מוצא את מקומו - גם חברתית. יש לנו יכולת לעזור לנווט את ההיכרות של ההתחלה, את ההזדמנויות ליצירת אינטראקציות חיוביות, שלא כוללות רק בקשת עזרה והדרכה אלא היכרות. התפקיד שלנו הוא לעזור לאנשים להשתלב, והחלק החברתי חשוב מאוד להצלחה של כולנו.

מתי יודעים שזה לא מתאים?

כפי שחשוב להציב יעדים ברורים לג'וניור החדש שלנו, גם לעצמנו חשוב להציב יעדים ברורים של עצמאות, דילבור וקצב למידה/עבודה. אנחנו מאוד ממליצים להגדיר את היעדים הללו בצורה ברורה ומדידה גם עבורנו, וגם עבור המפתח. בצורה הזאת נדע אם אנחנו לא פוגעים ביעדים. כשזה קורה באופן עקבי, ולאחר ניסיונות שונים לדרכי עבודה ועצות שונות, נצטרך להבין כי כנראה הארגון שלנו לא מתאים להתפתחות של אותו המפתח. **זה לא אומר שהוא איננו מתאים להיות מפתח!** רק שאנחנו, בתור מנהלים, או הארגון בו אנו נמצאים, פחות מתאימים לו. זה יכול להיות עקב גודל הארגון, הקצב המצופה בו, מורכבות המוצר והטכנולוגיה או המון גורמים אחרים. בסופו של דבר, חשוב לדעת לשחרר כשצריך, על מנת לא לפגוע בשאר הצוות ובהנאה והעתיד המקצועי של המפתח החדש שלנו.

איך לבנות תוכנית פיתוח אמיתית לצוותים טכנולוגים

TLDR: איך בונים מסלול התפתחות קריירה מקצועי למפתחים? לפי איזה פרמטרים מגדירים התפתחות? ומה צריך לקרות כדי לעבור ממקום אחד לאחר?

מושג התפתחות קריירה הוא נושא חם שיש עליו אינספור בלוגים ומאמרים. זה בעיקר מעלה שאלה סביב אוכלוסיית ה-IC, שההתפתחות שלהם לא מקודשת בטייטלים שונים וקידומים פומביים. כולנו יצורים מתפתחים, ושום דבר לא נשאר קבוע. אנחנו גדלים ומשתנים וזזים כל הזמן. אבל איך בונים את הדבר הזה בצורה מתודולוגית ככה שנוכל לנווט את הדבר הזה? בעינינו, בניית מסלול קריירה קצת דומה לספינה ששטה באוקיינוס. היא מתקדמת באמצעות הרבה כוחות: הרוחות, עוצמת הגלים, זרמים תת קרקעיים. אבל יש לה גם מנוע, ויש קפטן וצוות, שיודעים מתי לסובב את ההגה, או לפתוח מנועים ולעזור לנווט אותה קדימה לאן שאנחנו רוצים להגיע.

בניית מסלול קריירה בתוך החברה יכולה להתחיל ממש בכל שלב. בטיילור, יש לנו מסלול קריירה שהתחלנו לבנות כשהיינו 6-7 מפתחים. זה נשמע מוקדם, ואולי קצת קורפורייטי, אבל יש לזה הרבה סיבות.

בהרבה מקומות יש שיח בין המנהל למנוהל - במה אני מתפתח, מה הדבר הבא, מה אני צריך לעשות. העניין הזה של בהירות הוא קריטי. בהירות היא מושג סובייקטיבי וחמקמק כשהיא מגיעה לתחומים אישיים. וכמעט כל מנהל מכיר את הדיסוננס המתסכל שבו המנהל בטוח שהוא עושה המון, והמנוהל מרגיש שהוא לא זז מספיק.

בעולם ה-Engineering, שיש בו תנועה ערה בין חברות והרבה עיסוק של חברי צוות במיקום שלהם ובהתקדמות שלהם, אנשים רוצים לדעת אם הם ג'וניור, מיד, או סניור. וגם שהם מתפתחים ומשנים את המיקום שלהם לאורך הקריירה (ובצדק).

אנשים רוצים להתקדם, לגדול ולעשות יותר. יש חברות שהמיקום של מפתח משפיע על הדרגות בחברה, ואם זה בחברות שאין בהן דרגות, עדיין קיים מדרג של ותיקים לעומת חדשים.

ב-Domain Knowledge, ברמת המורכבות הטכנולוגית, בעבודה מול הסטייק הולדרים, בתחומי האחריות שמקבלים וגודל ומורכבות העשייה, גם אם הוא לא מוגדר בדרגות. אנחנו מאמינים שמסלול קריירה הוא לא סולם, אלא דרך. אתה לא מטפס - אתה הולך בדרך שהיא חשובה לא פחות מהיעד.

למה כדאי להוביל מהלך כזה בצורה ארגונית? זה מייצר הרבה מאוד בהירות ואחידות. וככל שיש הרבה צוותים - זה מייצר סטנדרטיזציה ביניהם. זה מונע מצבים שבהם בצוות אחד תהיה סניור ואילו בצוות אחר תהיה מיד.

המטרה היא לייצר ז'רגון פנימי אחיד (והוא נכון לתוך החברה, ולא דווקא מחוצה לה).

עוד נושא חשוב, הוא נושא הפרדות:

אם אנחנו רוצים לתת לאנשים להתמקד בפיתוח שלהם, כדאי לעשות הפרדות בין הבלים ל"מוקדי רעש". לדוגמא, טייטלים זה מוקד רעש משמעותי. גם קומפנסציה היא מוקד רעש. אם אנחנו רוצים לייצר מצב שבו אין לנו מרוץ פיקטיבי סביב הבלים, שאנשים יהיו במלחמה לעבור לבל (גם אם הם לא לגמרי שם), ולהימנע מפוליטיקות סביב הנושא הזה של קידום לבלים, כדי לשמר אנשים מהסיבות הלא נכונות - כדאי להפריד ביניהם. תוכנית הבלים נועדה לתמרץ התפתחות אישית, גדילה, אחריות ומורכבות עשייה. המוטיבציה להתקדם בבלים נותנת גושפנקא להתפתחות. מקצועית ההתפתחות הזו נותנת הרבה ביטחון והרחבת Skill set מקצועי. ככל שמתקדמים, העשייה היא יותר קשה ומאתגרת - וזה חלק מההתפתחות הזו. ככל שמתקדם מים בבלים, זה מאפשר יותר חופש בחירה. ככל שיש לך יכולת רבה יותר לקחת משימות שונות ומגוונות, אז גם האופציה שלך להשפיע על מה שאתה עושה גדלה ומתרחבת. כדאי גם לייצר Dual Track שבו יש שני מסלולים - מקצועי וניהולי. וכדאי גם לייצר גשרים שמאפשרים לאנשים לדלג בין המסלולים, ולייצר התפתחות שהיא לא רק לינארית, אם שם המוטיבציות שלהם ויש להם את היכולות.

אז איך זה נראה בפועל?

- בטיילור יש לנו שלושה לבלים - 1,2,3. לכל לבל יש את אותם 9 פרמטרים:
1. עצמאות ו-Ownership - המידה שבה המפתח לוקח אחריות על דברים ומוביל אותם באופן עצמאי. כמה הוא זקוק לעזרה כדי ללמוד דברים חדשים ולעשות אותם.
 2. איכויות של להיות חבר צוות - המידה שבה המפתח עוזר לשאר הצוות ומלמד אחרים.
 3. ידע וטכנולוגיות של הארגון - המידה שבה המפתח מכיר את המערכות. כמה Domain Knowledge יש למפתח?
 4. קואצ'ינג ומנטורינג לאחרים - המידה שבה המפתח מהווה מוקד ידע עבור חברי צוות אחרים, ומוודא שהידע שלו מתפשט בתוך הצוות והמחלקה.
 5. הבנה של הביזנס - כמה אני מבין את העסק ואת הלקוחות?
 6. חשיבה אסטרטגית - קדימה ולא רק על מה שקורה מחר.
 7. בניית צוות - מבחינת תרבות, גיוס, ותרומה לשימור של חברי צוות.
 8. Oncall - פרודקשיין אונרשיפ. איך אני מטפל בסביבת פרודקשן ומוודא שהיא תהיה למעלה? איך אני חונך מפתחים חדשים לתוך זה?
 9. Execution - ברמת המפתח, באיזה קצב, כמה אני מדייק בהערכות הזמנים שלי וכמובן - איכות הייצור.

הפרמטרים הללו נכונים לכולם, והם משתנים ברמת הציפייה, בהתאם לבלב.

אז איך זה עובד?

לב 1: המפתח יעבוד ברמת הפיצ'ר. הוא עובד עם שאר חברי הצוות, מתחיל להיכנס ל-Oncall, יודע ללמוד יחסית לבד, יודע לבקש עזרה ולא נתקע בבורות, ויש לו אחריות מסוימת על איזור (זה לא יהיה איזור מלא, אבל חלקים).

לב 2: המפתח יוביל פרויקט ולא רק פיצ'ר - Service או איזור מוצר שלם. כאן המפתח כבר מסוגל להעריך את המשימות שלו בעצמו ולהבין את המשמעות במאקרו, ולכן הוא גם יכול לייצר הערכות זמנים יחסית מדויקות. הוא גם יכול לקחת פרויקטים באזורים שהוא לא מכיר עם מעט מאוד הכוונה. הוא יודע לחקור בעיות קיימות, להביא פתרונות, עוזר למפתחים אחרים בצוות, ועצמאי לחלוטין ברמת Oncall - Tier 1. הוא יודע לתקשר בצורה ברורה עם סטייקהולדרים שהם לא טכנולוגיים כמו פרודקט או דיזיין לגבי החלטות שהוא מקבל. יהיה לו Domain Knowledge.

כל לבל עושה קואצ'ינג לבלב שמתחתי, ולכן פה, הוא יעשה קואצ'ינג למפתחים בלב 1.

לב 3: יודע להמיר רעיונות לפרויקטים עם משימות מוגדרות, שגם אחרים יכולים לקחת. יודע לשחרר חסימות של חברי צוות שלו ואולי אפילו של צוותים אחרים. אנשים באים לקבל ממנו תמיכה טכנולוגית ועזרה. הוא יודע לזהות איפה הוא זקוק לעזרה נוספת מארכיטקטים או Tech Leads וכו'. ההחלטות שהוא מקבל על ה-Tradeoffs בין חוב טכנולוגי, לבין התקדמות, הן מושכלות ונכונות. הוא יודע לתקשר עם חברי צוות לא טכנולוגיים בצורה ברורה. יש לו ניסיון ו-Track record של Ownership על האזורים שהוא עובד בהם. יש לו הבנה מעמיקה בארכיטקטורה של המוצר ושל השירותים והוא יכול להריץ POC, בדיקות היתכנות של דברים, והוא Oncall - Tier 2.

איך עוברים לבלים?

כדי לעבור לבלב הבא, אתה צריך לבצע את הבלב בצורה קונסיסטנטית לאורך זמן של כמה חודשים. עורכים שיחה עם המנהל, ושניכם צריכים להסכים שאתה שם באופן עקבי. וזה גם עוזר בארגון למנהלים לדעת מתי יש מישהו שהוא תקוע. כשעוברים על הפירוט של הבלבים, ומנהלים על זה שיחה קונקרטית, יכולה להיות חוסר הסכמה לגבי דברים מסוימים שעוד לא נעשו. למשל, חבר צוות יכול לומר שלא היתה לו הזדמנות לעשות משהו כדי להיות בלב 1. ואז חלק מהעניין הוא לא רק ללמוד, אלא גם להכווין, כדי ליצור מפתח יותר שלם. היות והתוכנית הזו שקופה לכל הארגון, היא מייצרת המון בהירות. וכולם מכירים את הבלבים אחד של השני. הפתיחות מאפשרת לומר דברים מאוד פרקטיים. למשל, יכולה להיות לנו סוגייה שאנחנו לא נגיד עליה שלבל 1 לא יכול לעשות אותה, אלא נאמר שלבל 3 צריך לאשר את זה. ובגלל שאנחנו עובדים בתהליך של קוד ריווי pull requests, זה מאפשר לפתוח את הדברים, ולתת ליותר אנשים אחריות.

מה קורה כשמגיע מפתח חדש?

כשמגיע מפתח חדש, הוא לא נכנס באף לבל, הוא בעצם מקבל גרייס (תקופת הסתגלות של בין חודש עד שלושה) כדי לסמן יחד את הלב שלו בהתאם לאיך שהוא עומד במציאות. בתהליך הראיונות מתרשמים, ורואים משהו, וההחלטה המשותפת צריכה להעשות ביחד עם המנהל. כל המפתחים שמצטרפים (וכמובן הקיימים) מכירים את הכל. כל המסגרת הזו נועדה לשרת אותנו ואת המטרות של הארגון וחברי הצוות. אנחנו לא משרתים את הכלי, אם צריך - נשבור את החוקים. זה נועד לייצר מבנה, אבל זה לא מחייב בצורה שהיא בלתי ניתנת לשינוי (אולי זה ההבדל בין סטארט אפ לבין קורפורייט). למנהלים יש שיקול דעת. הוא חשוב, ולשם כך התכנסנו גם למדריך הזה. אם המנהל חושב אחרת, ורוצה לשבור את החוקים, הוא יביא את זה לצוות הנהלה של הטק, והם יכולים להחליט יחד שמשנים את זה ושמים מישהו בלבל מסויים כפי שרואים אותו, אפילו אם חסר לו פרמטר אחד.

מתערבבים קצת עם פידבק

תוכנית לבלים עוזרת מאוד לייצר שיחות פידבק מאוד קונקרטיות. אם יש לנו פרמטר ברשימה, שמישהו אומר שהוא עושה אותו ומישהו אחר אומר שלא - מצויין. נוצרה פה ההזדמנות לשיח. התוכנית נותנת כלי מאוד חזק למנהלים, וכך גם למנהלים היותר צעירים, וגם למפתחים נוצרת הזדמנות לייצר שיח פידבק ברור. זה מייצר לכל חברי הצוות בהירות גדולה ומסלול התפתחות שהוא מאוד אפקטיבי. זה נותן הרבה מוטיבציה לקחת יותר אחריות, להוביל לפעילויות יותר מורכבות, ולא פחות חשוב - למנטר אחרים ולבנות את מערכות היחסים המקצועיות עם שאר חברי הצוות.

ניהול של פרודקשן - A Must

אין אקשן יותר טוב מסביבת פרודקשיין חיה ובועטת. בחברת B2C זה בכלל מלהיב. כל דבר שקורה מגיע מיידית ללקוחות. אתה יכול לדחוף קוד, וכמה שעות אחרי זה לראות אנשים משתמשים במה שבנית עבורם. זו חוויה מאוד עוצמתית, כי יש בה משוב כמעט מיידית לכל דבר שאתה עושה. יש אצלנו מיליוני מבקרים שעוברים במערכות שלנו כל חודש, ומאות אלפי לקוחות משלמים שמשתמשים בכל פיצ'ר קטן שאנחנו בונים - אפשר להבין למה נושא הפרודקשן הוא חתיכה מרכזית בניהול.

פרודקשיין רץ, והרבה פעמים יש תפיסות שגויות שאנחנו יכולים לתת לו לרוץ מבלי שנצטרך לנהל אותו. בהרבה חברות קטנות-בינוניות, פשוט משחררים את הנושא. לא מקצים לזה זמן, משאבים או אנשים. מתקנים באגים וזהו. בחברות גדולות, בדרך כלל, נמצאים צוותי SRE - שייעודיים לטפל בנושא.

בדרך כלל, בחברות קטנות, יש מי שמנהל את הפרודקשיין (אם אתם האנשים האלה, זה הולך להדהד לכם מאוד חזק, ואם זה לא אתם - השורות הבאות לא יראו לכם מוכרות). מי שמנהל את הפרודקשיין זה הבנאדם הזה שמתקשרים אליו כשיש בעיה. שמעלים אותו ישר על הקו כששמם לב שמהו לא תקין, בין אם זה ממערכת ניטור, חבר צוות שראה משהו במקרה, או לקוח לא מרוצה. אז למי יתקשרו? לבנאדם שמכיר את המערכות מצויין, שהכי זמין, שיכול לעלות על הבעיה הכי מהר, ולקדם את הפיתרון שלה בצורה הכי יעילה. בדרך כלל, מדובר באנשים הכי חזקים שלנו. וזו גם הסיבה הראשונה שצריך לנהל פרודקשיין, בכל חברה, ללא קשר לתלות. בפרקטיקה, במקומות שזה לא מנוהל, אנחנו שוחקים את האנשים שלנו. ובדרך כלל, יהיה מדובר באנשים הכי חזקים שלנו. התסכול מהשחיקה הזו מוביל גם אחרי זה לאנשים טובים שעוזבים אותנו. אפשר לזהות את זה בקלות גם אצל אנשים בראיונות.

הסיבה השניה לנהל פרודקשיין, היא שאם אנחנו לא מקצים לזה משאבים ביום-יום, אנחנו רק בתגובה ריאקטיבית למה שעולה מהשטח. אם אנחנו לא יודעים מה הפרקטיקה לביצוע כשיש תקלה, אנחנו כל פעם מחדש צריכים להחליט מה לעשות. כשיש פרקטיקות מוסדרות מראש, וכולם יודעים מה עושים במצב של תקלה, ההתנהלות יותר תקינה וחוסכת זמן. קצת כמו פיקוד העורף, שעורך תרגילים לשעת חירום ומנחה תהליכים סדורים למצבים שהם יותר קיצוניים.

הסיבה השלישית, ואולי הכי חשובה, היא שאנחנו רוצים ליצור אמון אצל הלקוחות, וכמובן ולפני הכל, אצל הארגון, ביכולת שלנו לזהות, לתחקר ולפתור בעיות. אם אנחנו לא מתייחסים בצורה גלויה לזה שיהיו בעיות בפרודקשיין, אז מהעניין כשיש בעיות, נוצר משבר אמון בין הפרודקט לאנג'נירנינג, בין ה-Sales לאנג'נירנינג, ובין ה-Account Managers לאנג'נירנינג.

במצב של תקלה, חווית אי הוודאות של הסטייקהולדים גדולה, וכשאינן פרקטיקה, אז גם אם תיקנו את הבעיה - אין שום ביטחון שהבעיה הזו לא תחזור על עצמה. ואז הסטייק הולדרים חושבים לעצמם: "אם אני זה שמעדיכן על הבעיות - אני לא יודע אם האנג'ניירינג יעלו על בעיות אם אני לא אציף אותן". ויותר מטריד מהכל - מה שאנחנו לא יודעים.

אז אחרי 3 סיבות טובות, ו"אני מאמין" שאנחנו מעדיפים לנהל דברים מאשר שהם ינהלו אותנו, יש לנו כמה מטרות כשאנחנו ניגשים למשימה של ניהול פרודקשיון:

- 1. לייצר אמון** - אנחנו האנג'ניירינג על הבעיות, אנחנו יודעים לטפל בהן.
- 2. בהירות** - אם קורה משהו, הסטייק הולדרים שלנו צריכים לדעת עם מי לדבר.
- 3. חלוקת עבודה** - עם מי מדברים ומתי (גם במחלקת הפיתוח בינה לבין עצמה) לצמצם שחיקה של חברי צוות
- 4. סדרי עדיפויות** - הבנה שאם עכשיו מתקנים - אז אין זמן לעבוד על פיצ'רים אחרים
- 5. עבור מפתחים מקצועיים** - זה חלק מהעבודה. אנחנו הופכים אותם לגדולים יותר, הקוד שהם יכתבו יהיה טוב יותר והתקלות שלהן יהיו פחותות. אם הם לוקחים חלק בטיפול תקלות, זה יותר אפקטיבי מכל מערכת של QA.

שאלת השאלות - להעלות דברים לפרודקשיון ביום חמישי? כן או לא?

אנחנו חושבים שזה תלוי בשני גורמים:

(א) האם יש לנו ביטחון מאוד גבוה שאם יש לנו בעיה במערכת אנחנו יודעים לגלות אותה ולטפל בה?

(ב) האם יש לנו יכולת להעלות דברים בקלות ולהוריד דברים בקלות?

אם שני הגורמים האלה מתקיימים, אנחנו לא רואים סיבה שלא להעלות דברים בחמישי. אנחנו תוך חצי שעה נדע אם יש משהו שהוא לא תקין. מעבר לכך, לדחות דברים מחמישי לראשון מייצר הרבה עומס, ואז גם הסיכוי לבעיות עולה, והרבה פעמים, אנחנו לא נרצה לאבד יום עבודה.

צוותים ייעודיים - כן או לא?

צוותים ייעודיים מייצרים אתגרים משל עצמם: הם מייצרים מעמדות בתוך המחלקה, וגם הרבה פעמים אנשים מרגישים נוח להסיר אחריות בשלבים מוקדמים כי יש מי שיטפל בזה אחר כך. בגלל שאנחנו מאמינים בעצמאות, אוטונומיה ואחריות - אצלנו זה לא מתיישב.

קויאט - לא תמיד כל אחד יכול לעשות הכל. יכול להיות שיש אנשים שטובים במשהו מסוים, ולא יהיו טובים בהכל. בארגונים קטנים, ההמלצה היא שכולם יעשו את הכל. בארגונים שגדלים, זה כבר תלוי בצוות ולא חייב להיות גורף. תמיד יהיו לנו אנשים שהם ברמות ידע שונות, בטח כשאנחנו ממשיכים לפתח דברים, ולא כולם בונים את אותו הדבר, ומזה נובע שלא כולם יכולים לעשות את התפקיד הזה באותה הצורה. אם אנחנו עושים "תורות", ויש למשל שבוע-שבוע לצורך הדיון, האדם הכי מנוסה, הוא זה שיפתור את הבעיה הכי מהר. אנחנו מייצרים Tiers ומאפשרים לעשות אסקלציה. כדאי לבנות תהליך אונבורדינג ל-Oncall ולייצר לו שכבות, מי שיש לו את רמת הידע הכי נמוכה, הוא הראשון שיקבל את הטלפון, וזה אומר שהוא לא יקבל אסקלציות מאחרים.

מדובר בעולם גדול של אי וודאות. כשיש תקלה בפרודקשיון, זה יכול לקרות מהרבה סיבות: שינוי בקוד, מערכות שאנחנו מתחברים עליהם, ספקי ענן ועד למתקפות על ה-DNS העולמי. ויש גם מערכות שהושפעו מקרניים קוסמיות. (אמיתי - תראו פה). רמת אי הוודאות של "מה תהיה הבעיה" מאוד גדולה, ומתוקף כך, אנחנו לא יכולים לאמן את האנשים לטפל בכל הבעיות, או לייצר אוטומציה להכל. האנשים הכי פחות מנוסים בקו החזית, ילמדו כמה שיותר דברים, כי אין לנו דרך מושלמת ללמד אותם מלבד ניסיון. הפרדוקס - כשאתה מגיע למערכת יציבה - האנשים שלך לא לומדים יותר להתמודד עם בעיות. אנחנו עושים ששנים, דוקומנטציה, כלים אוטומטיים שמסבירים איך להסתכל ולזהות בעיות, גרפים שמזהים אנומליות. זה חושף את האנשים החדשים מהר מאוד לדברים בעולמות הארכי-טקטורה.

כל המפתחים צריכים להיות מעורבים בטיפול פרודקשיון.

התמודדות עם טיפול בבעיות פרודקשיון חייבת להיות קריטית ומעל הכל. כמו שמצפים ממפתח לא לישון כי יש תקלה, אי אפשר לצפות ממנו לעבוד על פיצ'רים חדשים. אז מה עושים? כל תקלה שקורית צריכה להיות מעודכנת בצורה פומבית - מה קרה, מה היה האפקט ומה עושים עם זה הלאה. אחד הדברים שקורים בארגון כשעושים את זה, הוא שהגורמים השונים שעובדים עם הצוות מרגישים שותפות ואמפתיה לבעיות, הרבה יותר מאשר כעס.

מוניטורינג הרבה פעמים מרעיש בפאלס פוזיטיב, והמלחמה בזה צריכה להיות מלחמת חורמה. הטיפול בפאלס פוזיטיב צריך להיות כמו הטיפול בבעיה אמיתית, כי אם לא נעשה את זה, כל האנשים ישחקו, והאמון במערכת ירד. קצת כמו הסרט ההוא ששודדים בו בנק, וכל פעם מפעילים את האזעקה ומקפיצים את השוטרים לסיור - כשרק בפעם החמישית או השישית באמת ביצעו את הפריצה. זה נותן הזדמנות לתקן את זה עבור הבנאדם הבא ונותן יותר אמון במערכת. כשאנחנו מצליחים למצוא ולפתור בעיה מסוג חדש (יענו רק החברה החזקים שהצליחו לפתור) זה השלב לתעד, לייצר זיכרון ארגוני לדבר הזה, ולייצר מוניטורינג שהיה מזהה את זה.

ככה אנחנו מייצרים את המוניטורים הכי איכותיים שלנו, כי אנחנו מסמנים את הדברים שבאמת קורים, ולא רק הדברים שאנחנו חושבים שיקרו.

פרדוקס - רוב המוניטורים שאני אשים הם על הדברים שאני חושב שיקרו. אם אני מצליח לחשוב על זה שהם יקרו - אני כנראה אנסה גם לטפל בהם מבעוד מועד. הדברים הלא צפויים. המוניטורים הקלאסיים כנראה יתנו סימפטום ולא את מקור הבעיה. מערכת היא אורגניזם חי, והיא שונה מחברה לחברה, גם אם היא עושה את אותו הדבר, וחייבים לבנות את זה לכל החברה.

עבודה עם סטייק הולדרס

TLDR: האקוסיסטם של אנשים טכנולוגיים, וכמה עצות סביב האנשים שחיים בו ואיך לעבוד איתם.

הארגון שלנו מורכב מצוותים רבים בעלי תחומי מומחיות שונים. וכולנו יחד מנסים להגיע לאותן מטרות. הנקודה הכביכול ברורה ופשוטה הזו, מייצרת בפרקטיקה המון קשיים. בסוף הצוותים/מחלקות הללו מורכבים מאנשים בעלי מניעים ומנגנוני מדידה שונים שלפעמים מתנגשים. אנחנו בטיילור ניסינו מספר שיטות להתאים את המטרות ושיטות המדידה, על מנת להקטין את הקושי הזה. לפני 3 שנים, החלטנו לעבור לעבוד במתכונת של קבוצות עבודה משותפות, שאחראית על חלקים שונים במוצר **end to end**.

בקבוצות העבודה שלנו (פודים, או כמו שקוראים להם אצלנו - ברנדים) יהיו צוותי פיתוח, מובילים טכניים, דאטה אנליסטים, מעצבים ואנשים שעבודתם מתמקדת ביוזרים. עבודת צוות היא ערך עליון אצלנו, ואנחנו מאמינים שאנשים שונים אחראים על חלקים שונים בשרשרת. אבל כשכולם עם מטרה אחת, זה מקל מאוד על ההתנגשויות שקורות באופן טבעי כשלכל אחד הצרכים שלו. אנשי פיתוח הם חלק אינטגרלי מהביזנס, והם חלק מיחידה אורגנית אחת שמקדמת מטרות. אנחנו מאמינים לגמרי ששותפות בעשייה של התוכנית צריכה לקרות במשותף ובשקיפות מלאה. כולם צריכים להכיר את כל הנתונים, את כל המספרים, ולדבר באותה שפה. אין מצב בחברת סטארטאפ שהניתוח הצליח, אבל החולה מת. זה או שכולנו הצלחנו או שכולנו הפסדנו, ואין מחלקה אחת "מנצחת" ומחלקה אחרת "מפסידה".

בין אם אתם עובדים בקבוצות כאלה, ובין אם אתם עובדים במחלקות מסורתיות, ננסה להתמקד פה בדפוסי התנהגות והתנהלות שיכולים להתאים לכל שיטה (בתקווה).

מטרת העבודה המשותפת

חשוב מאוד לבסס עם הצוות באופן קבוע וחוזר על עצמו, את הסיבות שבגללן אנו עובדים על מה שאנחנו עובדים עליו. מה אנחנו מנסים להשיג ברמת הארגון? ברמת הקבוצה? ברמת הצוות? כמה שנשקף טוב יותר לצוות שלנו את הסיבות שלשמן הם עובדים על המשימות, זה יקל עלינו בהפרדה בין עיקר לטפל, וביצירת אחריות משותפת להשגת המטרות בצורה הטובה והיעילה ביותר. אם מפתח מקבל משימה וכל מה שמדברים איתו עליה זה זמנים ואיכות, הוא באופן טבעי יכנס למגננה וינסה לbuffer את הזמנים שלו, על מנת לדאוג שלא יבואו אליו בטענות. הוא כל הזמן יחשוב מה יכול להשתבש ואיך הוא מגן על עצמו. לעומת זאת, אם נדבר על המטרה כגורם מרכזי, נוכל לייצר שיח שמעודד חשיבה יצירתית לקיצור זמנים, לא ממקום של לחץ, אלא ממקום של הגעה ליעד ולולויו כמה שיותר מהר. בנוסף נזכה בבנוס שבמפתח, שהוא זה שנמצא הכי קרוב לעשייה, יוכל להציע רעיונות ליעול ושיפור המשימה. זה יכול להיות בהקשר לתהליכי העבודה, או ממש לפיצ'ר המסויים.

תזכרו - אנשים יתנהגו בהתאם לאיך שנתייחס אליהם. כמה שנקטין אותם ונבודד אותם, או כמה שנגדיל ונערב אותם - כך הם יגיבו אלינו.

עבודה עם פרודקט

יש הרבה סוגים של סטייק הולדרים שונים שמפתחים עובדים מולם: פרודקט, עיצוב, אנליסטים, מכירות, תמיכה, ניהול לקוחות, מנכלים, ועוד ועוד. אבל הכי נפוץ, ובעבודה הכי מרובה ברוב הארגונים, הוא איש הפרודקט. לשם כך, נתייחס ספציפית אליו ואל סגנונות נפוצים שלהם.

תפקיד הפרודקט מנג'ר מגוון, רחב ומשתנה מאוד מחברה לחברה. עקב כך הוא מכיל סוגים רבים של רקעים מהם מגיעים אליו אנשים שונים. באופן כללי, הנקודות החשובות שנרצה לשים לב אליהן כשעובדים עם פרודקט, או מדריכים את הצוות שלנו לעבוד איתם הם:

1. תקשורת פתוחה

נרצה לומר את הדברים בכנות ובתכיפות גבוהות. נזכור כי אנשי הפרודקט הם חלק בלתי נפרד מהצוות שלנו. כמה שהתקשורת שלנו תהיה פתוחה יותר (לשני הצדדים), יתאפשר לנו לקבל החלטות טובות יותר. תקשורת טובה לא אומרת להיכנס לכל פרט ופרט: כמו שהפרודקט לא יכול להיות חלק מכל החלטה הכי קטנה שלנו, כך גם אנחנו לא יכולים להיות משלהם. תמונה שלמה ככל הניתן - תעזור לכולם. זה גם חשוב לייצר תקשורת ישירה ולא דרך צד ג', כמו שהרבה פעמים קורה כשמישהו בא לשתף על מקרה שקרה, והשאלה הראשונה שלנו כמנהלים תמיד צריכה להיות: דיברת איתו על זה ישירות? אמרת לו את זה? פתחתם את הנושא הזה? הרבה פעמים אנשים חוששים, ואנחנו יכולים לעזור להם לבנות את המסלול לייצר תקשורת ישירה שבה הם מדברים אם יש משהו שמפריע, במקום ללכת לפתור את זה דרך צד ג'. תקשורת פתוחה לא קורית סתם ככה, זו עבודה. זה חשוב לייצר את המופעים והפלטפורמות לתקשר, לעבוד על זה בשוטף, לדבר על איך אנחנו רוצים לתקשר ולבחון מה חסר לכל אחד מהצדדים בתקשורת היום, כדי להמשיך ולהשתפר בזה לאורך זמן.

2. בהירות בקבלת החלטות

כפי שאנו רוצים להבין את הבחירות בתעדוף המשימות והפיצ'רים, ואת הסיבות העסקיות המניעות אותנו, כך גם אנחנו רוצים לספק תמונה מלאה ובהירה ככל הניתן לגבי ההחלטות הטכניות אותן נקבל. מה הטרייד אופים שעשינו בהחלטה, מה זה יעזור/ימנע מאיתנו לעשות בעתיד, עד כמה אפשר למקבל למספר מפתחים את המשימה ולמה. כל זאת ועוד, נרצה **to over communicate** את מה שאנחנו עושים על מנת לייצר בהירות. שנעשה זאת, נקבל פחות פושבק בנוסף מכיוון שההיגיון שלנו יהיה ברור. ואפילו אם כן יגיעו שאלות, הן לא יהיו פשוט "אפשר לקצר?". אנחנו רוצים להגיע להערכות אמיתיות משני הצדדים וללמד את כולם לדבר בכנות על מה נדרש, במקום להוסיף לכל דבר עוד 20% כי אנחנו בעצם מכוונים לפחות. זה לא בריא.

3. אחריות

ככל שנראה אחריות גבוהה יותר על העשייה שלנו, כך נייצר מערכת יחסים מלאת ביטחון שתיתן מקום ליותר סיפוק ואווירה נעימה בעשייה, ופחות תחושה ש"בודקים אותי" או "מעירים לי".

Assume good will practice.4

אחד הדברים המעניינים בחברות קטנות הוא שיש רק אפשרות שכולם ניצחו או שכולם הפסידו. בניגוד לחברות גדולות, שמחלקה יכולה להתהדר במוצר נהדר שבנתה, ועמדה ביעדים וחגגה הצלחה אפילו שהמוצר לא עלה לפרודקשיין או שהפרויקט נסגר, בחברה קטנה הדבר הזה לא יקרה. יש מטרה אחת שכולנו סביבה, וההצלחות שלנו תלויות אחד בשני. אחד הערכים שלנו הוא Start with trust. יש לערך הזה הרבה מופעים ביומיום שלנו שקשורים באוטונומיה ובאופן שבו אנחנו מנהלים את החיים והניהול שלנו. אבל אחד המפתחות בערך הזה הוא להניח שהצד השני מגיע עם כוונות טובות. גם הוא, ממש כמוך, רוצה לראות את הדבר הזה מצליח. כשאנשים נכנסים לכל דיון וכל פגישה עם נקודת המוצא הזו - דברים טובים קורים. הדבר הזה נכון על אחת כמה וכמה במערכות יחסים שיכולים להיות בהם אינטרסים מתנגשים, אבל אם עולים עוד רמה למעלה - תמיד מגיעים למקום שבו אנחנו חולקים יחד מטרה של הצלחה משותפת. אנחנו מתרגלים וחוזרים כל הזמן על הדבר הזה כשאנחנו בשיחות שיש בהן אי הסכמות וקונפליקטים.

5.ענוה

כשהמחשבה שאתה יודע יותר טוב מהצד השני מפעפעת בך - היא תמיד תמצא דרך להתגנב החוצה באיזו אמירה או אפילו בשפת גוף. הצד השני קולט את זה. וזה הרבה פעמים מעורר את ההרגשה שיש מולי מישהו מתנשא. כשאנחנו מרגישים שמי שמולנו מתנשא - זה מייצר הרבה מרחק ומפריד את העשייה המשותפת. מצד שני, הענווה להבין שאנחנו עובדים על אותו הדבר ורוצים להגיע למטרה משותפת, והענווה להבין שאני לא יודע הכל - היא בסיס מצויין להתקרבות. הנקודה הקריטית פה היא להיות תמידית במצב אמיתי וכנה לחקור ולנסות להבין את נקודת מבטו וההשקפה של מי שנמצא מולנו. לנסות לגלות מה הוא יודע שאולי אנחנו לא ואיזה תפיסת עולם מישהו מחזיק בעקבות המידע שיש לו שגורמת לו לחשוב את מה שהוא חושב. החקירה המשותפת והרצון הכנה להבין את הצד השני, יחד עם הענווה שאני לא בהכרח יודע הכל - מייצרת תקשורת טובה והתקרבות. וזה חייב לקרות אצל שני הצדדים.

7.מחר יש עוד יום

אנחנו לא פה ליום וגם לא ליומיים. כל מערכת יחסים היא ארוכה ומתמשכת, ואנחנו חייבים לזכור שמה שנעשה היום, משפיע ובונה את מערכת היחסים של מחר. בין אם זו קלישאה של חכם או צודק, או קלישאה של לנצח בקרב או לנצח במלחמה, אנחנו חייבים לזכור שמי שאיתנו, יהיה שם גם מחר. ואנחנו רוצים כמה שיותר, שהמעשים והפעולות שלנו יהיו כאלה שלא רק ייטיבו איתנו היום, אלא ימשיכו לבנות בינינו קשר טוב גם מחר. הפרספקטיבה הזו חשובה, כי היא מכוונת את כולם לזה שאנחנו רוצים לבנות משהו יציב לאורך זמן, וכמו בכל קשר, כדי לייצר בסיס טוב אנחנו חייבים להתחשב בזה שמחר אנחנו נעבוד עם הבנאדם הזה. כמו הפרקטיקה במשא ומתן שאומרת שאף פעם אל תנצח כשהצד השני מושפל, (אלא אם לא תראה אותו יותר בחיים), גם כאן, צריך לזכור שאף קונפליקט לא נפתר טוב, אם יש צד שמרגיש שניצח וצד שמרגיש שהפסיד.

8.תחקור רציף

העבודה המשותפת שלנו לא תגמר, וכך גם שיפור התהליכים והתקשורת המשותפת. עלינו לשים מופעים קבועים (נניח אחת לרבעון) ולדבר על מה שעובד לנו טוב ומה שפחות, כדי שנוכל להשתפר. לפי רמת המורכבות, לפעמים כדאי להכניס לשיחה גם פסיליטטור שעוזר לנהל שיח מובנה כדי לבנות את התהליך הזה בצורה פרודקטיבית ועניינית כמה שיותר.

הם ואנחנו

אחת הרעות החולות בכל ארגון, היא התנהלות ושיח ה"הם ואנחנו". כלומר, נתייחס לעבודה כלא משותפת ובעצם "נשכח" שאנחנו ארגון אחד ולא מספר ארגונים שמעבירים ביניהם משימות. בטיילור אנחנו מאוד רגישים, אפילו ברמת השיח, לדיבור בסגנון של "הם ואנחנו". חשוב מאוד לזכור ולתזכר את הצוות כל הזמן, שאנחנו ביחד במשימות שלנו. זה נכון גם לגבי הפרודקט וגם לגבי כל שאר הצוותים. דוגמא אישית בנושא תעזור מאוד, ותראה למפתחים שלנו איך להתנהל באזורים בהם אנחנו רוצים לפנות להתנהגות הזאת באופן טבעי.

פוסט מורטם

TLDR: איך מנהלים פוסט מורטם (כולל דוגמאות וטמפלט פצצה!) לתהליך שאשכרה יכול לעזור לנו ללמוד ולהגיד את האמון בארגון?

אחד הערכים שלנו בארגון נקרא: **Try.Fail.Repeat**. זה לא ערך "באוויר", אלא ממש דרך חיים. יש לנו פלטפורמת עבודה שמאפשרת לנו לעשות הרבה נסיונות, מה שנקרא: "לבדוק את המים" ולראות אם משהו עובד או לא. על כל דבר אנחנו עושים A/B Testing, מרימים משהו חדש, וחושפים אותו רק לחלק מהמשתמשים. במצב כזה, אנחנו יכולים לדעת אם אנחנו רוצים להתקדם עם משהו או לא, לפי התוצאות של כל קבוצת משתמשים. בהגדרה, זה גם אומר שבהכרח, תמיד תהיה קבוצה שהצליחה לעומת קבוצה שנכשלה. זה מלמד אותך לראות את הכישלון כדרך חיים, ללמוד ממנו ולהתקדם. מספרים שאדיסון עשה 1000 נסיונות לפני שהצליח עם נורת החשמל. וואלה, גם אנחנו, בצורה אנושית מאוד, טועים המון בדרך.

הסיפור של טעויות קורה בתוך כל ארגון, יש ארגונים שבהם נורא מפחיד לטעות, יש כאלה שיש בהם סביב הדברים הללו המון שיח של אשמה. טעות היא חלק בריא מחברה שגדלה ומתפתחת, ויכולה להיות חלק מהתרבות הארגונית בלי חווית אשמה, רק כשיודעים להכיל אותה, ללמוד ממנה, ולהמשיך הלאה.

אז מה עושים כשטעינו בדרך?

לפני שנכנס לתהליכים ומטרות בואו נבסס שפה משותפת. הרבה פעמים מדברים על רטרו (רטרוספקטיב) ועל פוסט מורטם, ולא תמיד ברור מה ההבדל בין השניים, והאם בכלל יש כזה. אנחנו מצאנו שההבדל בשפה שימושי לייצג שתי מטרות שונות אך דומות. ההבדל בין רטרו ופוסט מורטם נוגע בעיקר להבדל בגורם שהביא אותנו לתהליך הזה.

רטרו הוא בחינה כללית של משהו שעבר. זה יכול להיות פרוייקט, ביקורת על תהליך חדש שהטמענו, או כל דבר בעצם. פוסט מורטם, משמש אותנו להבין יותר טוב תוצאה בלתי רצויה שהגענו אליה. ובדיוק רב יותר, נועד לעזור לנו ללמוד איך נוכל למנוע מהתוצאה הזאת לחזור על עצמה. על אף העובדה, שברמה העקרונית, כולנו מאמינים בתחקור כלל התוצאות (הרצויות והפחות). לא רק בסביבת העבודה אלא גם בחיים האמיתיים הנטייה הטבעית של רובנו היא להתמקד דווקא בתחקור והבנה/מניעה של תוצאות לא רצויות. עקב כך, אנו מוצאים כי פוסט מורטם הינו הרבה יותר נפוץ, ועל כן החלטנו להתעמק דווקא בערך הרב שיש בלעשות אותו נכון ויעיל.

מתי נעשה פוסט מורטם?

פוסט מורטומים לא נולדים סתם כך. הם בד"כ נובעים ממשהו שקרה. תהליך לא יעיל, פגיעה עסקית, פגיעה באיכות. הגורם המניע הוא לרוב החלק הפשוט ביותר בתהליך, כאשר אנחנו מגיעים לתוצאה שברור לנו שהיא לא רצויה, ויש לה השפעה מהותית כלשהי על הארגון או הלקוחות, נדע כי זו סיבה טובה לעשות פוסט מורטם.

הבסיס לכל פוסט מורטם הוא למידה.

קודם כל אנחנו רוצים ללמוד ולהבין מה היה. לא נוכל להשיג את המטרה שלנו ולמנוע ממשהו להישנות, אם לא נבין אותו לעומק. אנו מנסים לייצר כוונה ופעולה בתוך תהליכים כדי:
א. למנוע ממה שלא תורם לנו או מעכב אותנו מלהישנות.
ב. לוודא שמה שעוזר לנו שחשוב שיקרה - אכן יקרה.

מהם כללי הזהב לפוסט מורטם מוצלח?

כלל זהב ראשון: להגיע לתהליך כמה שיותר נקיים. זה יכול לקרות כשמנקים מהתהליך הזה, כמה שאפשר, את כל הרגשות החוצה. בשביל לעזור בזה, עושים תהליך מאוד מוגדר וקבוע. נקי אומר שאני מגיע שהמטרה שלי זהה לשל שאר החברים בתהליך - והיא להשתפר. לא למצוא אשמים ולא להוציא תסכולים.

כלל זהב שני: לנסות ולעשות את הפוסט מורטם כמה שיותר מהר וצמוד לאירוע שהטריג אותו, כדי שהמידע יהיה כמה שיותר טרי לכולם. אנשים פחות זוכרים ככל שהזמן עובר, אפילו אם נראה להם שכן, והאיכות של התוצר עולה ככל שעושים את התהליך קרוב יותר לאירוע. כמה מהר זה טוב? עד 48 שעות זה מצויין, מעל זה, כבר מתחילה לרדת האיכות, וזה כבר ממש ניכר.

כלל זהב שלישי: אנחנו רוצים לשתף בפוסט מורטם את כל מי שהיה חלק באירוע/גורם. הזווית שלנו של מה שקרה היא מצומצמת לתפיסת המציאות שלנו. יכול להיות שקרו דברים שלא היינו מודעים אליהם. כל אחד שהתערב והיה קשור לאירוע מכל מיני סיבות (טובות או לא טובות) צריך לקחת חלק בלמידה, כדי להסיק מסקנות אמיתיות, ואנחנו רוצים שכמה שיותר מהמעורבים ישתתפו.

כמה טיפים כלליים לפני שנכנס לתהליך לפרטי פרטים:

מהרגע שעוברים משלב העובדות, נכנסים לעולם של דיעות, זה אומר שמי שמוביל את תהליך הפוסט מורטם, לא חייב לקבל את מה שכולם אומרים. אבל הוא מאוד רוצה לשמוע. יש לזה ערך מוסף נוסף - לכל האנשים שמעורבים ולקחו בזה חלק היה גם אימפקט על התוצאה הסופית, ואתה רוצה לתת להם הזדמנות להשפיע גם על איך הדבר הזה ייראה בעתיד. ולכן, ככל שהם יותר מעורבים באירוע, הם גם יותר רוצים להיות חלק מהפתרון ולקחת אחריות קדימה.

מה גם, כשאנשים מעורבים בתהליך וכמובן גם בפיתרון, והם חלק ממנו, גם אם הם רואים שהפיתרון לא מושלם - זה מחבר אותם בסוף למחויבות קדימה למה שהוחלט. אנחנו קוראים לזה "להזיז את הוילון" - לקחת חלק גם בתהליך וגם בפיתרון.

אז איך עושים את זה?

לפגישה של פוסט מורטם תמיד יהיה owner. לרוב זה יהיה מי שהיה הכי מעורב, והוא מנהל את התהליך.

אפשרי גם שיהיה מנהל לתהליך שלא היה מעורב, או אפילו לא היה באירוע עצמו. זה מאוד תלוי בנסיבות (למשל, עד כמה מי שהיה הכי מעורב מספיק מנוסה ובוגר כדי להוביל תהליך כזה שהוא מצליח להישאר "נקי"). אנחנו תמיד נתחיל בשיחת הכנה לכל המשתתפים, בעיקר לכל מי שזו פעם ראשונה שהם משתתפים בתהליך כזה, ואת שיחת ההכנה יעביר מישהו שבקיא בתהליך.

איך נראה התהליך?

1. להבין מה קרה - מיפוי מלא ומדויק של כל מה שקרה, עד הפרט הקטן ביותר. זה תהליך עובדתי נטו, יבש, בלי הטיות. ביום שלישי ה-27 ליוני בשעה 18:11, נדב עשה x ובשעה 18:15 יעל ענתה לו y. הרבה פעמים נעשה את זה גם בגוגל דוקס שכולם ממלאים בבת אחת. אפשר להיעזר בסלאק, בג'ורה, טרלו, פייג'ר דיוטי, כל המערכות שמחזיקות מידע על השתלשלות האירועים, על מנת להוציא את המידע הכי מדויק. לא לנהוג בקנאות לרמת הדקה כאשר המידע לא זמין. כן לנהוג בקנאות לגבי סדר השתלשלות האירועים.

2. להבין מה היו האימפקטים - מה נגרם מהאירוע ומה היו התוצאות? מה זה עשה? איבדנו מכירות? פגענו באמון של היוזרים? בזבזנו שעות עבודה? להשתדל להגדיר תוצאות מבוססות נתונים, כי זה לא תמיד שחור ולבן. זה בסדר, ואפילו מומלץ, לצרף גרפים ודוחות רלוונטיים. גם חלק זה צריך להיות עובדתי ככל הניתן, ואמור לא לגרור התנגדויות או דיונים.

3. להבין מה היו הגורמים המרכזיים שגרמו לזה - פה זה יותר קשה מפני שלרוב יש פה מימד סובייקטיבי: חוסר ידע, חוסר תקשורת, טעות אנוש, התנהלות לא נכונה. ננסה להגיע לגורמים כמה שיותר מדויקים. במקום לומר "שנדב חדש בארגון" - נאמר "מה נדב לא ידע". כדי לרדת לשורש הבעיה, אפשר גם להשתמש בכל מיני שיטות של חתירה לגורמי שורש, 5y's יכולה להיות למשל שיטה טובה.

4. להעלות פתרונות מוצעים - כל פיתרון צריך להיות ישים (יעני - מחובר למציאות ובר ביצוע). אנחנו גם ננסה שפתרון יענה על כמה שיותר גורמים בבת אחת. ושהפתרונות יהיו כמה שיותר bulletproof. מה זה אומר? שאם הפיתרון מבוצע, אנחנו לא יכולים לחזור על הטעות או הבעיה שהוא בא לפתור. "צריך לזכור" זה לא טוב, כי אי אפשר לדאוג שמישהו יעשה את זה.

אולי נרצה לשים שלב שבו יש מישהו שצריך לאשר באופן ידני, איזשהו שלב חוסם - אי אפשר לעלות לפרודקשין בלי שעובר אישור של דודלי.

5. בקרה - אם הפיתרון שהצענו היה קורה, האם זה היה מונע? האם יכול להיות מצב שהפיתרון לא יקרה?

- + דוגמא לפתרונות שעולים בדרך כלל ולא עומדים בדרישות:
- + להעביר הדרכה
- + להוסיף משהו לאונבורדינג
- + לזכור משהו כשעושים משהו
- + לכתוב מסמך הסבר

פתרון טוב יהיה כזה שאם מיושם, הבעיה לא יכולה לקרות.

6. בחירה מתוך הפתרונות המוצעים - מה פותר הכי הרבה ומה בעלות הכי נמוכה (משאבים, זמן)?

7. ומה עושים כשאין פיתרון? - לא תמיד אפשר למצוא פתרון. יש מקומות נדירים שבהם אי אפשר למנוע הישנות. במצב כזה, יכול להיות שיש צורך בשינוי יותר עמוק, ולייצר תקשורת שבה אנחנו מבינים שהבעיה יכולה לחזור וצריך להוסיף ניטור. אולי אנחנו לא יכולים למנוע הישנות של מצב כזה, אבל אנחנו יכולים לצמצמם **Time to discovery or time to resolution**.

אם אנחנו בתהליך הזה מספר פעמים ומזהים בעיות ש"אין להן פתרון", זה הזדמנות טובה לערב אנשים אחרים מהארגון שאולי יוכלו להעלות רעיונות, או לשקול את צורת העבודה שלנו וחלוקת האחריות שגורמת למצבים הללו להישנות.

8. Action Items - לכל סעיף ברשימה צריך owner ותאריך יעד. מי שמוביל את התהליך יודא שיש את הכל, והוא אמון לראות שהכל קורה על ידי ה-owners השונים.

9. שיתוף - בטיילור נחשוף את התוצאות לפורום יותר מצמצם כדי לקבל פידבק. השיתוף הראשון הוא תמיד ב-Time box. למשל עד מחר אפשר לתת הערות, ואז מפרסמים לכל הארגון.

למה אנחנו תמיד מפרסמים לכל הארגון? קודם כל זה מייצר למידה משותפת. מכל פוסט מורטם אפשר ללמוד בכל הארגון. דבר שני, זה מצמצם את הבושה: "הנה מה שעשינו והנה מה שלמדנו, וזה בסדר". מצג ברור איך אנחנו עושים דברים. כאשר אנחנו משקפים לשאר הארגון את התהליך בצורה ברורה: ניתוח, לקיחת אחריות ומציאת פתרונות, אנחנו יכולים להסיט את השיח מעולם של האשמות והפניית אצבעות ללמידה והתפתחות. זה מייצר מעגל של ביטחון. זה יכול להיות המנכ"ל או צוות מקביל, או אנשי CS שצריכים לענות ללקוחות.

ונקודה אחרונה לסיום:

דוגמא אישית - לוודא שתמיד עושים את זה. מעל איזשהו סף - עושים פוסט מורטם. בכירים או לא, כולם עושים את אותו תהליך ומשחררים לכל הארגון. מראים שמדובר בתהליך בריא' ולא עניין של הסתרה. לחזרתיות יש משמעות/ כמה שעושים את זה יותר, מאמינים בזה יותר.

בעיות תמיד יקרו. אבל את הבעיה הזאת לא נחווה שוב. אנחנו מצמצמים את מרחב הבעיות שלנו. אחרי 2-3 כאלה, כבר מתחיל להיווצר השיח, שאם סטייק הולדר חווה משהו - **VP Product** או **VP Sales**, במקום לשאול מי אשם, הם ישאלו מתי יהיה פוסט מורטם. ואז למשל מסיימים את הטיפול ב- **Incident** ואומרים פוסט מורטם יצא עוד יומיים.

הדבר הכי חשוב - לצאת עם משהו שימנע את הבעיה או יפתור אותה מהר להבא. זה הופך את התהליך לאמיתי במקום לחוויה של "מראית עין" או איזו חותמת גומי.

בונוס קטן

כמו שבטח כבר הבנתם, אנחנו מאמינים שפידבק טוב הוא כזה שקורה עכשיו, באופן מייד. לכן, קיבלנו מזמן החלטה לוותר על תהליכים מסורתיים של הערכות ביצועים, שקורות בארגונים אחת לחצי שנה (במקרה הטוב), ואחת לשנה (במקרה הרגיל). הטפסים הארוכים האלה שממלאים בהם הישגים, חוזקות, תחומי פיתוח ושיפור, ו"השיחה החגיגית" שעושים המנהל וחבר הצוות יחד אחרי, הרגישו לנו מלאכותיים ולא מספיק מקדמים.

אז מה כן? החלטנו להקדיש את שיחות ה-1:1 השבועיות שממילא קורות אצלנו בארגון, לשיחות פידבק שבועיות.

שיחות ה-1:1 הן בעצם המונוי שעות שכל ארגון משקיע בהם, אבל אם תחשבו על זה, סביר להניח שאף אחד לא לימד אתכם או יישר אתכם קו על מה עושים בארגון בשיחות 1:1. יש את אלה שעושים את מה שהמנהל שלהם עשה - כי זה היה מעולה. יש את אלה שעושים בדיוק הפוך ממה שהמנהל שלהם עשה - כי זה היה על הפנים. ויש את אלה שהגדילו ראש, פתחו בלוג או קראו קצת חומרים והבינו לבד מה מתאים להם. אנחנו החלטנו שהזמן הזה הוא בדיוק המקום להתלבש עליו כדי באמת לעשות ongoing feedback הלכה למעשה. עכשיו רק נשאר לנו ללמוד יחד איך עושים את זה.

למה אנחנו מספרים לכם את כל זה?

כי בין היתר בתוך הלמידה גם חיפשנו כלי שיעזור לנו (בכל החברה) בשיחות פידבק שבועיות, כלי שהוא פאן וקולע, משהו שיסייע להקליל שיחה של שעה פעם בשבוע, או לפתוח ערוץ שדרכו אפשר לצלול פנימה דרך מטאפורה ולגרד קצת את פני השטח הרגילים של "איך הולך?". בסופו של דבר, בנינו את הכלי הזה בעצמנו - יצרנו לכל חדר ישיבות חפיסת קלפים פיזית, שיש בה 90 conversation starters לשיחות פידבק (ובעולם היברידי, איך לא, גם יצרנו גרסה וירטואלית למשחק הקלפים).

אתם יכולים למצוא אותו פה ולנסות בעצמכם - הוא פתוח לחלוטין.

ואגב, הוא גם עובד גם עם ילדי בית ספר שאוהבים לומר שהיה היום "סבבה" / בערב זוגי/ בארוחות משפחתיות.

תהנו!

Post mortem

Date	
Participants	

Title

A description of the incident

Incident Date

Start date and end date

Event summary

A summary of the event

Users' impact

How were our users affected

Company impact

How Tailor Brands was affected

Timeline

Date	What happened
dd/mm/yy	

Issues

Main issues that we can note from the incident

Suggested Solutions

What can we do differently to avoid the issues from repeating?

Suggested Solutions

What can we do differently to avoid the issues from repeating?

Action items

Action item	Owner	Ticket link	Due date	Comments