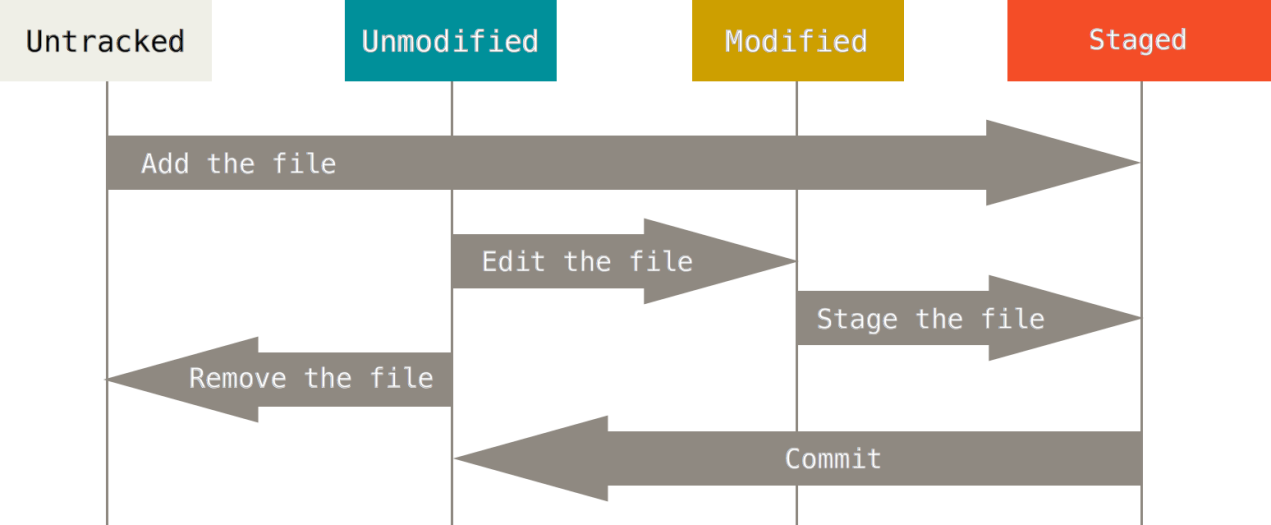
החלון cmd מאפשר הרבה דברים, כשמתחילים משהו בgit זה הוגדר לך בעזרת ההתקנה של git להפעיל את ההמשך דרך הgit, בגלל זה כל פקודה שתמצא ברחבי האינטרנט שהיא פקודה של git תצטרך לכתוב לפניה git.

נתחיל בלדבר על תיקיית גיט במחשב שלך, בלי קשר לשרת וכאלה.

הקונספט של גיט עובד ככה שעוקבים אחרי כל קובץ בנפרד, הגיט זוכר עבור כל קובץ את כל הגרסאות שלו.

איך קובץ משנה גרסא?

ככה נראה התהליך:



Untracked – קבצים חדשים שלא היו בגרסא האחרונה, כדי להוסיף קבצים כאלה לStaged משתמשים בפקודה  
git add -A

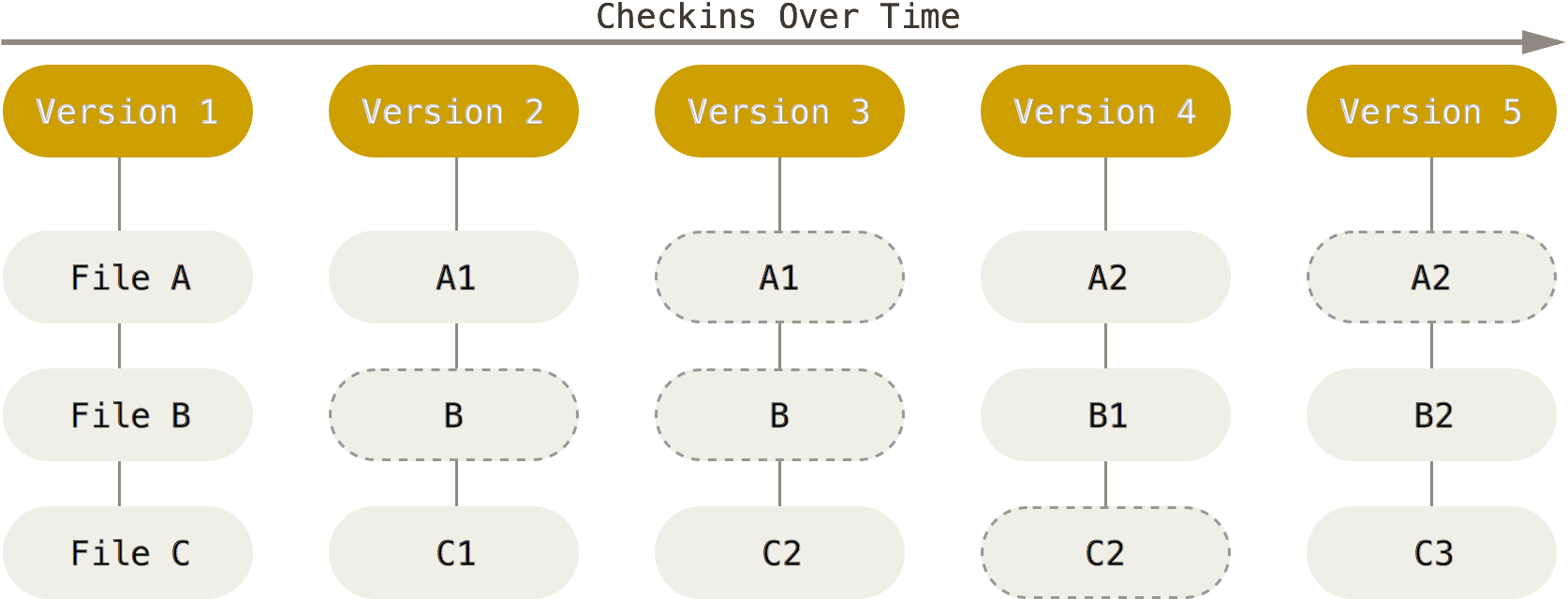
זה מוסיף את כל הקבצים החדשים, אפשר לכתוב שם ספציפי או סוג ספציפי של קובץ כדי להוסיף רק דברים מסויימים, כשתתעמק תבדוק על זה כרגע זה לא רלוונטי.

Unmodified – קבצים שכבר במעקב (כלומר היו בגרסא האחרונה) אבל לא השתנו, כדי להעביר אותם לModified פשוט צריך לערוך אותם (יענו לתכנת).

Modified – קבצים שכבר במעקב והשתנו מאז הגרסא האחרונה, כדי להוסיף אותם לstage כותבים  
git add .  
(שים לב לנקודה – היא אומרת להוסיף את כל הקבצים, שוב אפשר להכניס רק חלק אבל...)  
הפקודה שהצגתי להכניס קבצים שהם untracked מכניסה כאילו כבר כוללת בתוכה את הפקודה git add . אז אם גם ערכת קבצים וגם יצרת קבצים אתה לא צריך לעשות את שתיהן.

אז מה זה הstaged הזה בכלל ולמה שאני אכניס לשם קבצים?

זה בעצם הקבצים שיעודכנו בגרסא החדשה שתיווצר בעזרת הפקודה git commit –m "Your comment here"  
כשמתמשים בcommit לכל הקבצים שהיו בstage נוצרת גרסא חדשה לפי המצב שבו הם נכנסו לstage. הפקודה הזו בעצם יוצרת גרסא הבאה לפרוייקט, שבה מתעדכנים כל הקבצים שהיו בstaged, קבצים שלא יהיו בstaged ישארו בגרסא הזו זהים לגרסא הקודמת, הנה תמונה שממחישה את העניין:



(גרסא היא בעצם אוסף של פויינטרים, לכל קובץ פויינטר לגרסא המתאימה שלו בגרסא הכללית של הפרוייקט)

git status פקודה נפלאה שמציגה לך את המצב, איזה קבצים untracked ואיזה modified.

אז למה כל זה טוב אם זה עוד לא קשור עדיין לסרבר ועבודה משותפת?

אתה יכול בעזרת זה לחזור גרסאות אחורה, וגם להתפצל לענפים שזה סיפור אחר שתקרא בעצמך באתר של גיט בהזדמנות, הענפים זה כלי מצויין.

עכשיו לסיבה שלשמה התכנסנו כאן היום – עבודה משותפת.

זו תורה שלמה לעומת מה שאני עומד להסביר.

כדי לדחוף את הגרסא האחרונה שאצלך (כלומר הגרסא האחרונה שעברה commit) אתה כותב:

git push https//:github.com/DrJunk/PitchIT.git master

כדי לחסוך לך את כתיבת הURL את יכול להשתמש פעם אחת בפקודה:

git remote add <short name> https//:github.com/DrJunk/PitchIT.git

ומאז להשתמש בקיצור במקום בכתובת.

לפעמים הוא יכתוב לך שהpush נדחה על ידי הגיט, זה יקרה אם אני אעשה push מאז הpush האחרון שלך, ואז הוא ירצה שתשלב בין הגרסאות לפני שתדחוף גרסא חדשה.

כדי לשלב בין הגרסאות אתה מתחיל בלמשוך את הגרסא של הremote (מה שעל השרת) ככה:

git fetch <the short name or the url we just talked about>

זה לא עושה כלום לפרוייקט שעל המחשב שלך בינתיים, אבל זה מושך את העדכונים מהשרת.

אחרי זה אתה כותב את הפקודה הבאה כדי לשלב בין הפרוייקטים:

git merge <the remote name>/master

אם עבדתי על קבצים שונים מאלו שאתה עבדת עליהם זה יעבור חלק, יעדכן לך מה שצריך וזהו.

אם עבדנו על אותם קבצים הוא יגיד שיש conflict בקבצים מסויימים, תצטרך ללכת לקבצים האלה והם עכשיו יכילו גם את מה שהיה אצלך וגם מה שאני עדכנתי, ואז צריך פשוט לשלב אותם ידנית, ולעשות שוב את הפקודה של הmerge.

אחרי שסיימת לשלב בין הפרוייקטים אתה יכול לעשות push, אלא אם אני מניאק ועשיתי עוד push בזמן הזה.

זהו.