



Software craftsmanship – kvalitet, reflektion och lärande

Transkribering: Avsnitt 49

Programledare 1: Paula Jonsson

Programledare 2: Olivia Hellbom

Gäst: Jennifer Torquato



Transkribering

Paula

Så nej. Men hej Olivia.

Olivia

Hej Paula.

Paula

Tjena.

Olivia

Tjena.

Paula

Nu är det fredag och vi ska spela in pong.

Olivia

Ja äntligen.

Paula

Det var ett tag sen nu.

Olivia

Skönt att vara tillbaka och? God fortsättning till alla som lyssnar. Vi sa ju inte det för avsnittet eller ge det här blir ju jätte sent. Nej, jo, men du, det blir det tredje tredje månaden på året.

Paula

Ja.

Olivia

Men, vi har ju haft lite. Ja, vi har.

Paula

Ju haft lite avbrott.

Olivia

Jag.

Paula

Gick ju och. Red omkull, så man fick ju spendera de två första månaderna med brutet ben och OS, så då blev podden lite på pant.

Olivia

Väldigt bra tajming.

Paula

Det var jättebra tajming, det är det. Det har varit jättekul.

Olivia

Hade du planerat eller hur?

Paula

Folk hade ju trott det om jag inte var så ointresserad.

Olivia

Innan, nej, men sen jag är inte heller jätte OS intresserad, men i år så har jag kollat. Det har varit kul faktiskt.

Paula

Jo, men det har det och roligt med så mycket svenska framgångar och liksom. Bara börja med det här mix dubbel guldet med syskonen brano. Det var så himla fantastiskt alltså. Jag var jag var så engagerad. Jag satt och live uppdaterade min kille med SMS när han var iväg och spelade brändspel i Sverige.

Olivia

Ja men vad ja. Där har vi i alla fall anledningen till att vi har varit borta.

Paula

Exakt.

Olivia

Men vi är tillbaks nu och vi kommer köra på som vanligt och den här gästen kommer ju passa väldigt bra in tror jag. För att nu är liksom även om vi är lite efter. Vi är på månad tre redan och många har säkert redan liksom haft sin nystart på året liksom och sin så här. Man får ju ändå någon nytändning också. När det är nytt år känner jag, jag blir ju verkligen. Vi behöver gå in i mötet.

Paula

Ja.

Olivia

Hej och välkommen till det. 49 avsnittet av podcasten vadå i t tjej? Mitt namn är orde vid hellbom.

Paula

Och jag heter Paula Jonsson.

Olivia

Och idag så har vi med oss en gäst som heter Jennifer Torquato. Välkommen Jennifer.

Jennifer

Tack, så mycket kul att vara här.

Olivia

Jättekul att ha dig här.

Paula

Verkligen?

Olivia

Men jag tycker vi är jättenyfika på dig. Lite först om du vill berätta om ja, om dig och din. Bakgrund.

Jennifer

Ja absolut absolut jag efter. Ja, jag vet inte hur långt bak jag ska börja, men jag är född och uppvuxen i Stockholm. Jag. Pluggade på universitet i London International business så ingenting i t relaterat och efter när jag var nyexaminerad så sökte jag massa jobb på på måfå och alla möjliga jobb som verkar intressanta i London då. Och jag fick jobb på accenture som det hette Consulting analys och där fick jag prova på väldigt många olika olika roller. Jag jobbar testade en liten stund. Jag var business analys ett par månader och sen kom jag in på i t projektledning. Som jag tyckte om och som jag kände att ja, det här var min grej, så jag jobbade som i t projektledare några under några år i London och sen när jag flyttade tillbaka till Stockholm så började jag jobba på särskilt i var där fem eller sex år som service delivery manager. Och nu nu bor jag i Spanien i södra Spanien, nära Malaga och jobbar på cojorrence som också är ett i t konsultföretag. Som har som sin unik selling point softwarecraft manship och det är det som vi ska prata om mer idag det.

Paula

Är ju jättehäftigt. Vad har det alltid varit en självklarhet med med i t branschen eller hur kom du in på den delen?

Jennifer

Nej, det var av en slump faktiskt att när jag började på exempel så var det både strategi och i T Och ja mer bara hamnade in på fler och fler i t projekt och tyckte att det var kul och såg. Märkte det snabbt att det här är en bransch som utvecklas hela tiden? Det kommer det. Det känns som en trygg karriär för att i T Och teknologi och allt det går liksom. Väldigt mycket framåt och väldigt fort framåt och man måste hänga med i svängarna. Nu är det a i överallt och hela tiden och så där så jag såg det som en. En bra chans att komma in hyfsat tidigt. Det här var 15 år sedan, ungefär som jag började jobba så ja det. Det blev en slump och även congerence och det här att jag har lärt mig mer om software craft manship. Det var också en slump, men jag kände direkt att när jag hittade dit att det jag jag delar alla värderingar och arbetssätt så det klickade verkligen direkt så jag. Jag har haft tur med det där att ja, jag har helt enkelt hittat rätt.

Olivia

Gud vad spännande? Jo jo, men nämnde det lite, men vad alltså så här? Vad var det som som du fastnade med med liksom software crafip?

Jennifer

Så för att bara att ta det i korthet. Vi kommer gå in på djupet på alla olika delar men att förklara det bara lite lite kort till en början så software craft manship är som utvecklings hantverk eller? Det det handlar i princip om att man ska se utvecklande av kod som ett hantverk, så istället för att bara skriva den koden man ska göra och nöja sig så handlar det mycket om om kvalitet framför kvantitet. Det ska vara lätt att underhålla. Man ska ju skriva ren kod som är lätt att förstå, lätt att testa, lätt att ändra så att vem som helst i teamet kan komma in och och fortsätta. Det ska inte vara svårt eller komplext att att förstå. Sen har vi lärande att man det hör också till hantverk biten att man vill alltid försöka sträva efter att bli bättre, lära sig nya tekniker så att slutprodukten blir bättre i högre kvalitet och enklare att underhålla framåt också. Och sen är det att man ska vara professionell att ha ett engagemang för att skapa värde, inte bara leverera kod, men att det ska vara något som faktiskt fungerar och. Ger värde till till den kunden eller den de man jobbar för. Och så slutligen är det mentorskap och community att man vill dela med sig av den kunskapen man tar in i den lärande punkten för att höja hela branschens kompetensnivå att. Att, det är ja att man själv vill lära sig och så sen det man lär sig det som man tycker funkar bra vill man fortsätta lära ut till andra, så det blir som en ringa på vattnet att alla hjälps åt och det det är värderingar och arbetssätt som jag. Tycker det är väldigt bra.

Olivia

Så det är det är typ som ett ett arbetssätt, ett framework liksom.

Jennifer

Ja.

Olivia

Spännande.

Jennifer

Så det är det är man skulle kunna säga att. Alla software crafts people som vi kallar det är utvecklare. Eller ja, förlåt jag. Jag är ju inte det då eftersom jag är inte min roll. Jag vet inte om jag sa det i början, men min roll. Jag är engagemang manager nu som är det är lite likt service deliver manager att jag. Jag har hand om kundrelationerna med med våra kunder. För att se till att teamen våra team har det bra att leverera det de ska och att kunden är nöjd och glad och den processen så jag. Jag själv är inte super teknisk. Min mina starkare sidor är mer softskills, men det är också vi kommer till det sen när vi pratar mer om community delen och lärande att dela med sig att jag kan lära ut. Mer soft skills till utvecklarna och utvecklande de som vill och utvecklarna kan lära mig mer om de tekniska bitarna, så jag får mer förståelse för hur hur allting fungerar.

Olivia

Ja, men jag fattar för det. Det blir ju alltså, jag menar, det är ju agilt arbete det. Det inkluderar ju alla, även inte bara liksom utvecklarna utan alla i teamen. Och ja, så det är ju intressant att man kan få höra lite olika perspektiv på det.

Jennifer

Absolut och så när du säger teamen i just softwarecraft manship också. En viktig. Är att man ska äga hela processen. Från början till slut, så dels att man kan lite om allt som det vi kallar fullständigt man. Man förstår hela processen och skulle kunna. Vara med i alla steg från krav till design och utveckling, till testande och till produktion och så både fronten, backen, databas och allt möjligt. Det betyder inte att man ska vara specialist i alla ämnen. Det är ju omöjligt, men att man ser det som en pyramid eller som ett. Är kanske att man kan lite om allt och så sen har man sin specialisering. Det som man tycker är roligast eller där man har sina starkaste sidor. Men det gör att teamen fungerar väldigt bra. Att om någon är sjuk eller bortrest eller så så är det alltid någon som kan hoppa in och hjälpa till. Att man, det blir väldigt få botten next, säger man hals.

Paula

Flaskhalsar.

Jennifer

Flaskhalsar.

Paula

Nu, nu har du ju gått in ganska mycket på software craft manship, men när kom du först? I kontakt med begreppet och innebörden.

Jennifer

Det var när jag hittade till currence där jag är nu för att vår VD och en av grundarna till företaget är också en av. Grundarna och en av de liksom huvudnyckel personerna i software och craftmanship community över över hela världen. Han startade det i London och det är nu den största communityn som finns i världen. Men det finns många i olika länder tyvärr eller jag såg att det finns en i Stockholm och en i Göteborg. Men de ser inte ut att vara superaktiva, så om det är någon här som lyssnar så får ni gärna dra igång ett community. Men jo så vår VD då Sandro och mancuso. Han startade det här med sin kollega också som startade conturace han. Mars bado de ja Sandro skrev en bok eller han har skrivit en bok som heter the softwarecraft person som jag verkligen rekommenderar den. Den är lättläst för både tekniska personer och icke tekniska personer, så han har skrivit nästan. Det är väldigt enkelt att. Att förstå alla koncept och jag läste på om det är väldigt mycket när jag skulle söka jobb där för att jag insåg väldigt snabbt att jag ville jobba där och då, då behövde jag ju förbereda mig inför alla intervjuer och och sånt. Och ja, på den vägen är det jag har varit där nu i tre år och jag älskar det. Det kan ta en lite tid att komma in i det. Just allting som handlar om att ja. Man, man ska dela med sig om eller man ska dela allt man kan och så ska man vara väldigt nyfiken och vilja lära sig saker. Jag har alltid tyckt att det har varit kul att lära mig saker så det kom. Hyfsat hyfsat naturligt, men det handlar också om ge och ta mycket feedback och det behöver man också öva på för att man behöver rätt teknik för att det ska landa på rätt sätt. Inte bara klaga på någon utan det att ge feedback ska ju handla om att man vill hjälpa en person bli bättre, inte att man vill. Planka ner på någon bara för att så att man själv känner sig bättre eller någonting sånt där, så ja det, det var så jag igen mitt liv, det bara slumpar sig men men det blir så bra så. Så är det.

Paula

Det är lite typiskt i den här podden. Det är många som säger det att jag planerade att göra något helt annat från början. Men sen råkade jag halka in på på IT spåret och så helt plötsligt så hamnade jag här. Så det är. Det är väldigt intressant med att i t ibland bara händer.

Olivia

Jag satt och funderade på den här boken som du tipsar om om man om man jobbar liksom som utvecklare eller inom i t då och inte för jag antar att. Egentligen teorin eller liksom tanken är väl att man ska som team, då kanske allihopa jobba så här, men man kanske får ut någonting av att läsa boken för sig själv också. Kan man liksom typ. Det kanske låter jättekonstigt, men man kanske kan tänka på det sättet själv eller?

Jennifer

Ja. Definitivt så allting är applicerbart på sitt egna arbete, men det blir alltid bättre om man om man kan jobba i i samma stil i hela teamet. Men det beror ju också på hur. Hur upplägget är hur man jobbar tillsammans i team, skriver man kod i samma kode. Arkiv kodbas eller gör alla sin egna grej allt sånt spelar in. Men jag skulle absolut säga som som du sa Olivia. Börjar man själv att lära sig så kan man testa lite och så kan man kanske dela med sig av vissa insikter som man lär sig till sitt team ska man säga hörni, ska vi testa det här? Ska vi testa och köra och parprogrammera till exempel en gång och se hur det känns och. Och om jag jag jag gjorde, jag testade det här och det funkade jättebra ville att jag ska visa så kanske det liksom kan man kan rulla igång snöbollen som blir sen större och större och så sen haka fler på och det är oftast så det fungerar när man. Man hittar något som fungerade och vill, då blir man taggad och så vill man att andra ska förstå hur bra det är och så. Ja, blir det ringa på vattnet?

Olivia

Jo, men jag blir så nyfiken då för du nämnde ju. Nu har jag inte det framför mig exakt, men någonting i stil med typ att alla utvecklar inte är craft people. Eller kraft men.

Jennifer

Eller kraft, men kraft woman kanske både och. Nej, precis så software craft people är ju utvecklare, men alla utvecklare är inte software craft people. För det krävs ju då att man ska. Använda sig av teknikerna och värderingarna som är in inom software craft med ensipef. Så just det här att man ja dels jobbar agilt med extrem programming. Som test driven development parprogrammering snabba korta cyklar att Pusha till produktion små små bitar i taget så att man kan få snabb feedback automatiserade tester. Den typen av metodik. Samtidigt som man inkorporerar just värderingarna. Man ska vara pragmatisk och professionell även om jag jag är övertygad om att alla skulle vilja kallas professionella, så det det ordet kanske liksom inte riktigt blir landar. Jag vill inte att det ska landa för er, men även just det här att hela tiden lära sig hålla sig uppdaterad att testa våga testa nya nya metoder, tekniker, nya teknologier som man kanske inte. Det här ska från början, men man måste testa någonstans för att kunna börja lära sig och så kan man se ja, det här funkade för mig eller det här funkade inte testa någonting annat och och gå vidare från det, och det är så man man ständigt blir bättre. Man lär sig själv man lär andra för att man vill att. Man vill inte vara en, vi kallar det för lonely Rockstar att man vill inte vara den förr förr i tiden ansågs det bra att vara the go to person, att man måste komma till mig, för det är bara jag som har kompetensen att kunna lösa det här. Ja, men då har du ju en flaskhals om den personen blir långtidssjuk får till exempel eller byter jobb. Jaha, då sitter du där och så är det ingen som kan fortsätta arbetet för att det är så så svårt att förstå så enkelhet. Ödmjukhet och att vara pragmatisk i hur man jobbar är liksom huvudstenar i i konceptet.

Olivia

Varför alltså det här låter ju alltså otroligt? Ja. Jo, men alltså fast.

Jennifer

Det låter också jag tycker du när jag själv nu sitter och pratar om det. Det låter som att det borde vara så självklart, men det är det inte alltså. Tänk dig det. Det är jättemånga som sitter och kör sitt race och liksom är nöjda med ja, men. Jag tog den här kursen eller jag pluggade till utvecklare för 20 år sedan och nu kan jag det. Nej. Tänkte 1, 1, 1 utvecklare från från tio 15 20 år sedan om de skulle fortsätta utveckla nu. Som de gjorde då, de skulle inte få

ett jobb. Man måste hela tiden hänga med för att ha att ha en fördel i arbetslivet i sin karriär och för att kunna ge värde till leverera värde i form av. Både man skriver att det ska vara hög kvalitet. Det ska vara enkelt att förstå ändra en till bit i just extrem programming refactoring att man hela tiden ska. Ju ju mer man bygger på en kod ju ju längre den blir, desto mer komplexare blir den och det skapar tängens guld hög kostnad för underhåll. Den typen av av saker mer buggar ofta, men om man har som princip att varje gång man. Man skriver ny kod som gör en längre så ska man också försöka hitta någonstans där man kan omstrukturera lite, göra lite refactoring för att göra någon bit lite enklare att förstå så att den är en sån här scoutregel att man ska lämna. Man ska lämna det så som man hittade det eller bättre.

Paula

Men du? Du pratade om att du ville verkligen ta anställning på Codurance. Vad var anledningen till att det var just dit du ville? Var hade du sett dem någonstans eller hade du hört talas om dem eller var det just? Om att du hade hört om software craftsmanship som var grejen.

Jennifer

Nej, det var ja vi skulle flytta till Spanien, jag och min familj och då började och jag ville jobba remote. Och då och nu pratar jag flytande spanska, men då pratade jag mediumen inte. Jag kände mig inte så självsäker i att kunna jobba på på på spanska, så jag letade remote jobb i Spanien för engelsktalande. Och då? Ja, för det första fanns det inte jättemånga öppningar för det, men så hittade jag till cojorrence och redan i i arbetsbeskrivningen. Så kunde jag känna att det här jag klickar med det här tänket med det här med de här värderingarna och hur de jobbar och sen direkt i intervjuprocessen med de olika intervjuerna och testerna som vi gjorde. Där det där hör också det. Jag tror att det finns. Det är ett kapitel i sandrews bok också som handlar just om intervjuteknik och hur man skriver hur man skriver arbets beskrivningar när man söker personal, att det det handlar också om att man ska få ut det bästa av. Där man intervjuar igen. Jag har varit på intervjuer där folk vill testa ens stress eller nervositet, så sitter de och typ stirrar på en eller vill inte. Det är inte en konversation utan de ställer en fråga och så är det liksom. Man får absolut ingen känsla för att. Vill ni anställa mig eller försöker ni bara sätta mig på plats eller hur fungerar det? Men det var hela tiden en väldigt liksom trevlig process att man kändes även om man alltid är nervös när man söker nytt jobb så kändes det väldigt. Bra att de de vill att de vill förstå hur man tänker hur man resonerar mer än det här är ett svar, ja eller nej, typ av grej och när vi när vi anställer utvecklare till exempel så. Vad använder vi alla? Vi använder? Testerna handlar om extrem programming, metodik och intervjuer är pair programming, så då har vi de som jobbar för oss alltså de softroc people alla i i vårt företag. Alltså cross people och då är alltid en intervju en parprogrammering med en eller två från oss som som programmerar tillsammans med med den som vill bli anställd. Mer för att se hur hur de jobbar, hur de tänker, hur de skriver kod och det. Det får man ju liksom. Det blir inte bättre än så för att kunna få en känsla för om den personen är rätt för jobbet eller inte.

Paula

Men är det då speciella uppgifter som ni har med er, eller?

Jennifer

Är det vi brukar använda oss av? Det kallas katta. Jag vet inte om det är om det används mycket mycket i i Sverige, men här i alla fall så använder de mycket katas som är en kodövning. Att man sätter upp ett scenario och så. Ska man hitta en lösning och koda den biten på problemet och det kan ta en två timmar. Ja, vissa såna katter ska ta en hel dag och då kan man göra det som mob programming, att man gör det i grupp om man vill öva på en speciell teknik till exempel. Det är det är någonting som många soffor craft people gör inte bara intervjuer. Eller eller på jobbet, men mer som för sitt eget lärande. Det är en en inlärningsteknik som funkar bra för att då då tränar man liksom mer praktiskt på det man man har lärt sig eller vill lära sig.

Paula

För. Gör ni det då både på mer seniora eller är det mest juniora allihopa? Vad får ni för respons på just era sätt att intervjua på?

Jennifer

Väldigt bra, väldigt bra respons. Folk känner sig bekväma och. Just det här med att alla får ha sin personliga, individuella touch alla är olika. Alla jobbar på olika sätt, men i den här processen så så syns. Alla alla erfarenheter, alla alla färdigheter, allt som man kan. Det är väldigt lätt att se hur det fungerar i i de situationerna så. Man vill inte att alla ska vara likadana, man vill inte att alla ska följa samma mönster så länge koncepten, värderingarna metodiken är där och just den här viljan att vilja lära sig och vilja utvecklas. Hela tiden för det är otroligt viktigt. Då spelar det inte så stor roll om man inte kan en bit man kan till och med man man kan hamna någonstans och säga, oj, jag, jag vet inte riktigt vart jag ska ta vägen här att då våga fråga vad skulle du göra eller kan du? Jag behöver lite hjälp här för att kunna ta mig vidare. Det, det är mycket bättre att vara öppen med det än att försöka gömma sig, göra en snabb a i a i search eller googling och bara och bara vad ska jag göra här för att sedan man ser väldigt tydligt om det är någonting som någon inte kan som den försöker gömma? Så hellre att man är öppen med det, men att man säger jag har intervjuat folk mer på säljsidan eller managementsidan och. Då spelar det ingen roll. Jag säger inte nej till en person om den säger. Just det här har jag ingen erfarenhet av, men om de säger jag har ingen erfarenhet, men jag är säker på att jag kan lära mig det snabbt. OK då är det så, eller då är det så. Förhoppningsvis så är det så. Klart om man tar dem på orden?

Paula

Och.

Jennifer

Känslan.

Olivia

Kan alltså den här strävan för att vara bättre och eller förbättras? Och ja, men att lära sig och så vidare kan det liksom slå över på att det blir en liksom för mycket? Ja men stresspress och liksom jag vet inte att man.

Jennifer

Det kanske kan det, men det handlar lite om personligt ansvar och prioritering. Så det är i vissa fall om man är superstressad och har en jättetajt deadline på några veckor. Då kanske inte just det är de veckorna man ska fokusera på att lära sig någonting helt nytt som man aldrig har testat tidigare. Men jag tror att det är bra att kunna få in så rytmen att precis som man behöver prioritera för att man måste försöka hitta tid i veckan. Att röra lite på sig så får man försöka hitta lite tid att lära sig någonting, läsa någon artikel, dela någonting. Det finns ofta finns det events på linkedin eller meetup eller så som man kanske vill lyssna in på medan man eller poddar eller det finns ju egentligen hur mycket information som helst om allt möjligt så så länge man är intresserad så så kan man hitta det mesta. Och så sen skulle jag väl säga att man behöver inte eller behöver man ska väl. Jag skulle inte rekommendera att man väntar tills man känner sig supersäker och har läst tio böcker om ämnet eller så där utan säg att du läser en bok, antingen om soffor chefsminiship eller jag gillar mycket böcker om softy. Till som teamwork management, ledarskap, feedback och kommunikation och så och så fort jag har läst ett kapitel eller några sidor så försöker jag tänka, hur kan jag träna eller applicera det här? På någon situation, alltså mycket av softskills, till exempel övar jag på med mina barn sen någon någon violent Communication att hur kan man? Hur ska man säga någonting för att få det man vill? Hur var det som funkar på kunder kan även funka på barn, men man måste ju såklart vara ja ändra ordvalen lite kanske, men koncepten är detsamma så likadant om man börjar

lära sig om extrem programming till exempel så fort du har börjat lära dig. Gå in på github. Det finns massa såna här öppna open source Project heter det öppna käll projekt brukar.

Paula

Man inte bara säga open source. Det är sopen source, ja.

Jennifer

Ja. Open source Project att man kan liksom gå in och titta på vad andra jobbar med. Kan man bidra lite, testa eller föreslå för en kollega eller kompis? Ska vi parprogrammera lite så kanske vi kan lära oss av varandra. Det gör man ju ofta när man ser hur någon annan gör så jaha, så där hade inte jag tänkt göra om jag hade. Du sa att jag gjort så där, och så börjar diskussionen aha intressant. OK, men vi testar och ser min det eller mitt förslag eller ditt förslag och kan vi förbättra här? Kan vi lösa det här problemet på något sätt? Så just att hitta små. Små moment i i både under arbetsdagen, men också utanför att men men intresset måste ju vara där att man vill lära sig inte att ska man tvinga sig att lära sig så då blir det inget kul och då förmodligen inte så hållbart heller.

Paula

Nej precis. Men du nämner ordet extreme programming. Vad vad innebär det för någon som aldrig har hört det förut?

Jennifer

Så det skapades av en man software eller en utvecklare som heter kent beck. Han har också skrivit en bok som heter extrem programming expanded om man är intresserad av att veta mer och det handlar i grund och botten om att eller fokuset ligger på enkelhet samarbete. Snabb feedback. Korta cykler att man ska Pusha till produktion ofta och i små steg så att man hela tiden kan gå tillbaka och ändra om det behövs. Och tre de tre större grenarna är test, driv and development, TDD parprogrammering och refactoring eller omstrukturering. Så om jag ska förklara de tre också lite snabbt så TDD test streamen development. Handlar om att man ska skriva ett test innan man skriver koden för att då det tvingar utvecklaren att förstå kravet innan de börjar koda och då kallas det röd grön refactor, så det är tre steg att man ska skriva ett misslyckat test som är rätt. Sen ska man skriva minimal kod så lite som möjligt för att det ska passera till grönt och sen ska man gå tillbaka en sista koll och se kan jag göra den här kod biten lite bättre, en liten refactory. Och det här kan det kan ta lite tid för en nybörjare att komma in i det och det kan vara lätt att tänka att äsch ja. Jag skriver min kod och så skickar jag det vidare till 1 1 1 testare så får de ta hand om det, men det är det är när man väl har lärt sig. Tidigare det så är det väldigt effektivt. Det är mycket högre kvalitet. Färdig buggar. Det är lättare att fortsätta jobba, det är automatiserade tester så det finns test på allt det är. Det är mycket bättre. Det blir mycket bättre dokumentation. Det sparar tid och pengar för det blir färre buggar, mindre kodunderhåll. Och ja, det är det är när man väl har. Lärt sig tekniken så har jag inte hört någon som inte rekommenderar och sen med ett par programmering så. Är det handlar det om att man är två stycken helt enkelt som som jobbar i i samma kod? Det är en som skriver koden och en medan man skriver kod så förklarar man hur man tänker vad man gör och vad man planerar att göra. Och sen den kallas för driver och så sen är det den andra som är navigator eller granskare som ser koden och då har man, då använder man ju då fyra ögon istället för två så att den som ja båda kan. Hitta fel eller den som är navigator kan också komma med förslag. Precis så som jag nämnde nyss att, jaha, du gjorde så där, OK? Jag tror att det blir bättre om man gör på det här andra sättet. Jag tror att det kan det eller oj titta där du gjorde ett litet misstag som skulle kunna bli en bugg så vi löser det direkt nu så det blir en feedback. Session plus att man lär sig av varandra för att man ser man ser hur en jobbar och man hör hur den andra skulle göra så. Diskussionen som kommer upp i parprogrammering. Det är både både lärande och. Och där motiverande. Det kan dock ta också lite tid att komma in i det så man behöver träna och det är även individuellt hur mycket man vill och orkar programmera. Jag har vissa kollegor som kan sitta hela dagarna med varandra och och och programmera samtidigt som de såklart. Snick snackar om om lite annat ibland och andra som hellre vill gå in och.. Parprogrammerat. OK, vi jobbar lite själva och så sen kör vi en timme

eller två och. Vi kör mer som en code reve att nu tittar vi på de här bitarna tillsammans eller här är jag lite osäker på om jag har om jag har gjort det bästa eller har du någon feedback på det här till exempel och sen den sista då refactoring eller omstrukturering att tror att jag nämnde det? Tidigare också, men att man ju ju längre koden blir desto mer komplext blir det desto svårare blir det att underhålla och det är lättare eller ja? Till slut så får man ju massa teknisk skuld och så därför ju. Det handlar inte om att man ska omstrukturera allt hela tiden, men att man har det hela tiden i bakhuvudet att då och då gå tillbaka kan man förenkla någonting här och där att göra det lite enklare lättare att förstå och lättare att läsa lättare att fortsätta jobba med så gör de det. Det är värt. Hela tiden det tar för att det ger mycket framöver. Det skapar det. Vad säger man man minimerar eller reducerar problematiken som kan komma längre fram om man om man inte om man inte gör det?

Olivia

Så. Jag, jag vill ju fråga typ ta det ta det längre tid att att jobba för min min bild är att sitter man och parprogrammerare som du sa, så kanske man inte är lika produktiv, men det är ju absolut liksom säkert en felaktig bild.

Jennifer

Ja, jag skulle säga. Säg det för att det är så först och främst, man måste ju komma in i det så man måste träna för att kunna det är lätt att i början att man kanske känner sig blyg att man inte vill prata så mycket och. Det är väldigt viktigt att man gör det för att om man inte delar med sig och pratar om det man gör och delar feedback och olika åsikter och idéer på hur man hur man går vidare då då förlorar du ju poängen. Men det den lilla den extra tiden det tar att vara två. Det får du igen nästan alltid i högre kvalitet och färre buggar. Får du buggar så kommer du ju behöva gå tillbaka och spendera tid på att lösa dem så då då kommer den tiden där istället så dels så så. Ja, så tjänar du in på det med färre buggar och högre kvalitet om du parprogrammerar plus att du får lärandet att man delar kunskaper med varandra så man kan hela tiden utvecklas på det sättet också. Så det är som en liksom en bonus.

Olivia

Hur förhåller det sig till alltså nu när a i n kommer in i bilden för att programmerare och eller kommer in? Det har ju varit ett tag, men sitter man då tillsammans och fortfarande? Liksom ja.

Jennifer

Det där är så det finns ju nu mycket a i agenter och lösningar som att man kan ha programmera med. Med AI eller att man kan ha a i som hjälper en att ge feedback på sin kod eller att a i n skriver koden och man själv ger feedback och det är ju såklart ett alternativ. Men jag skulle säga att det blir lite mindre lärande där. Det finns allt fortfarande risk för att a i n gör fel och det är inte lika roligt. Det blir ju inte lika socialt, så är man inte social så kanske det funkar bättre. Men de flesta tycker att det är trevligare och roligare att prata med en kollega än med AI.

Olivia

Men.

Jennifer

Man kan göra båda.

Olivia

Ja en annan del då feedback och mentorskap. Det är jag jättenyfiken på. Kan du inte berätta om det.

Jennifer

Så. Just så tillbaka till bara snabb tillbaka på extreme programming så handlar det om att man ska göra små. Man ska Pusha små kolbitar till produktion för att få snabb feedback tillbaka. Sen är det. Feedback i det större perspektivet att man vill lära sig och man vill utveckla så man vill bli bättre. Man vill finslipa sina tekniker, lära sig nya tekniker för att hela tiden bli bättre och det bästa sättet är att få feedback. Så man får komma in i rytmen att hela tiden eller ja i i parprogrammering också. Då får man ju konstant feedback mellan varandra när man jobbar jobbar tillsammans. Men också att efter en presentation eller efter något möte det kan vara små saker, så säg att du har ett möte med en kund när ni diskuterar någonting. Prata med din kollega som var i det i i det mötet också. Ja, har du någon feedback? Hur tyckte du att det gick? Sa jag sa jag något. Någonting knäppt eller något som du inte hör med om, eller hur presenterade jag mig eller någonting sånt och även försöka komma in i rytmen att. Ge feedback, hej du, jag noterade det här. Det behöver inte vara stora grejer. Ibland är det stora grejer och då behöver man formulera sig väldigt, väldigt noggrant för att det ska landa väl, men även små saker igen. Så vi på mitt jobb, så vi har både. Kanaler på Vi använder slack chattfunktionen och vi har olika grupper och så där och då har vi en kuros eller ja kulos kanal där man lägger upp framför alla. Och så taggar man någon person och så säger man ja, den här det, det är ju en positiv feedback. Man säger alltså inget konstruktiv feedback, men för det handlar ju om båda det. Det är jättekul att ge positiv feedback och det är väldigt lätt att. Ignorera det att inte göra det för att man tänker att ja, men allt man ska göra bra ifrån sig för att det är det som förväntas av en. Men det är alltid väldigt kul att få positiv feedback, så vi försöker ofta liksom få igång att alla ska ge varandra positiv feedback i den här kanalen till exempel och. Ja den här personen har levererat den här funktionen till den här kunden och de blev jättegglada. De sa det här och det här, så det vill vi dela med alla och det är roligt och så blir det massa emojis och alla kommer med och liksom bra jobba jättebra. Men sen med konstruktiv feedback som ju då är lite mer icke kan vara lite mer lite svårare att dela med sig av så dels så är det lättare om man öppnar upp genom att fråga själv, har du någon feedback till mig? Men om det är något som jag ser som någon gör och den inte ber om det, men jag tror att den personen skulle behöva ändra på den biten, då får man bara. Öva försöka försöka säga det på ett så konstruktivt sätt som möjligt här. Vi har också haft många möten och Sessions om att ja, men hur man hur man levererar feedback och hur man lyssnar på feedback. Som jag nämnde non violent Communication att man berättar så här känner jag nej eller så här tänker jag när du gör så här att man inte ska använda sig av, du gör alltid så här eller du gör aldrig det där att man ska liksom bara så specifik man kan. Men och även ge ett förslag. Om jag vore du, så skulle jag som jag nämnde tidigare att jag ömmar när jag pratar att om jag var du skulle jag försöka öva på att istället för att säga så var tyst när du är klar att prata för att feedback i grund och botten handlar om att man ska. Bli bättre man ska lära eller lära sig något nytt eller sluta göra någonting som kanske skadar någon annan eller som någon annan kan ta illa upp eller som inte är så bra för projektet man jobbar med så man borde vilja. Höra den feedbacken som andra har om varandra. Vi har även gjort lite så här, role play scenario så att man OK, det här har hänt. Du är den som har gjort det och du person b är du ska behöver berätta för den här personen. Hur skulle du göra det? Och använd till exempel nonvianal Communication principen att hur hur man liksom formulerar sig på rätt sätt och. Ja, jag jag ja feedback är är bra.

Olivia

Men skapar man en miljö där man ständigt ger och tar feedback både positivt och negativt så tänker jag att man allihopa också blir mottagliga för för det.

Jennifer

Absolut så det handlar om. Ja, men förmodligen börja smått igen och där hjälper det mycket om till exempel managers och team leads. De personerna börjar så är det lättare för alltså. Det kan alltid vara svårt för någon junior att ge. Feedback till en senior men det är det man behöver försöka jobba bort att, men man måste börja någonstans och då är det lättare för för leda det. Ledarteamet att börja att både ge positivt och konstruktiv kritik eller feedback i olika olika situationer, små saker, stora saker, det som det som behövs och gärna så här. När det när det kommer till positivt så är det ju alltid bra eller då? I principen när det är positiv feedback involvera alla alltså publika kanaler eller det teammöte. Säg det framför folk. Före Paula, du gjorde det där riktigt snyggt bra

jobbat, då blir man glad och konstruktiv feedback ska man ta 1, 1, 1 och en för det handlar bara om den. Den personens utveckling och behöver inte höra hemma hos någon annan.

Paula

Tycker det låter som en väldigt bra idé, speciellt det där med kanalen med positivt. Det blir ju, det blir ju bara en massa positiva saker vi kan skriva om varandra och det behöver ju inte vara så himla stort. Det utan det kan vara minsta minsta lilla sak typ så här idag spelar jag in ett avsnitt med Olivia och hon var helt grym. Jag är så glad att få göra det med henne alltså bara såna saker.

Jennifer

Jag håller med.

Paula

Jag ser att klockan börjar trilla iväg klockan är 12:52 i alla fall här. Jag vet inte. Har vi samma tidszon? Nej.

Jennifer

Jo, vi har samma tid.

Olivia

Är det någonting som du känner förresten innan som du vi har missat nu?

Jennifer

Nej, alltså jag. Jag skulle säkert kunna prata en timme till, men vi får se kanske några gånger i framtiden kan vi köra en falop med att gå lite djupare i vissa punkter? Men nej, jag tror. Jag tror jag har fått med mig eller fått ut det det det viktigaste och jag hoppas att några kanske blir lite nyfikna och vill lära sig mer om softwarecraft manship. Så det finns ja, förutom sandros bok som heter the crafts the software craftsman. Så finns det. Jag tror det är en hemsida som heter softwarecraftors.org som visar alla olika community som finns i världen och. Där kan man gå med i någon om man vill det finns det är event det finns på linkedin också eller den här kent becks bok om XP som heter extreme programming explain också intressant? Och ja, jag finns på linkedin om någon vill om någon vill connecta och snacka lite.

Paula

Absolut, det var faktiskt min sista fråga om om man kände sig lite nyfiken på vilket jag verkligen hoppas att man gör på software crowdmanchip vad man kan vända sig. Men jag tänker att vi lägger med de här länkarna som du har beskrivit i beskrivningen för avsnittet också så. Och är det enkelt att hitta rätt och jag tänkte detsamma att det här skulle vi kunna prata en hel timme till om vi skulle kunna göra en en del ett och en del två. Jag tycker nog att det hade varit roligt att bjuda in det i framtida avsnitt. Vem vet man kanske man kanske skulle kunna ta med någon som är lite nyfiken på? Hur ska vi kunna implementera det här i vårt företag och att man gör någon liten dialog om det?

Jennifer

Det vore jätteintressant. Räkna med mig.

Paula

Jajamän. Men tack så jättemycket för att du ville medverka i podden. Jennifer, det har varit.

Jennifer

Upp med det här, det var jättekul. Det är min podddebüt.

Paula

Det.

Olivia

Gick jättebra. Ja det var.

Jennifer

Roligt.

Paula

Ja, men det där var ju jätteintressant, Olivia.

Olivia

Verkligen?

Paula

Det var kändes som att det var en del ett och sen att det kommer en del två.

Olivia

Om man vill fortsätta.

Paula

Göra.

Olivia

Mer och lära sig för det känns som att man kan verkligen applicera det där på. Man kan verkligen applicera det här på på sin. I sin vardag oavsett vilken roll man är i och även då på sig själv som vi pratade om också att så här att man behöver inte ha ett team, utan du kan också sträva efter att lära dig mer eller dela din kunskap eller söka kunskap hos andra eller ja, alltså hela tiden. Så det här var att det blir bättre typ.

Paula

Ja men precis. Jag tänker that's a wrap

Paula

Så tack till dig som har lyssnat och har du någon idé på en person som du tror skulle kunna passa i podden eller har du feedback till oss i? I podden så får du jättegärna skriva till ad herr- podcast, snabelasoriety.com eller till mig eller Olivia på linkedin hejdå.