Gestione Tecnica Dello Sviluppo di Software

Esperienze, Consigli ed Idee Alex Martelli

http://www.aleax.it/it_tmsd.pdf



©2007 Google -- aleax@google.com

C'e` la Pallottola d'Argento?

...un singolo modo di far fuori tutti i mostri?







NON Parlo Di...

- (per nulla) gestione strategica/esecutiva
 - business plan, finanza, visione strategica...
- (poco o nulla) project management
 - o pianificazione, tempistiche, budget
- (appena appena) tecnologie/metodi/strumenti
 - o linguaggi, sistemi operativi, framework...
- ø do per scontato un <u>certo</u> livello di Agilita`:
 - ne` un rigido Waterfall,
 - ne`il totale Caos!-)



Agile vs Waterfall vs Chaos

- Waterfall: mira, mira, mira, mira, FUOCO!
- Ochaos: fuoco!, fuoco!, fuoco!, OOPS, ...
- Agile: mira, fuoco!, aggiusta la mira; mira, fuoco!, aggiusta la mira; ...
 - i progetti SW di successo devono avere una qualche misura di questa agilita';
 - onelle "metodologie agili" si cerca di cogliere sistematicamente gli aspetti che funzionano, e applicarli con disciplina, adottando le tecniche di codifica e test che si attagliano alle esigenze reali



L'Agile, come le DP...:

The Timeless Way of Building

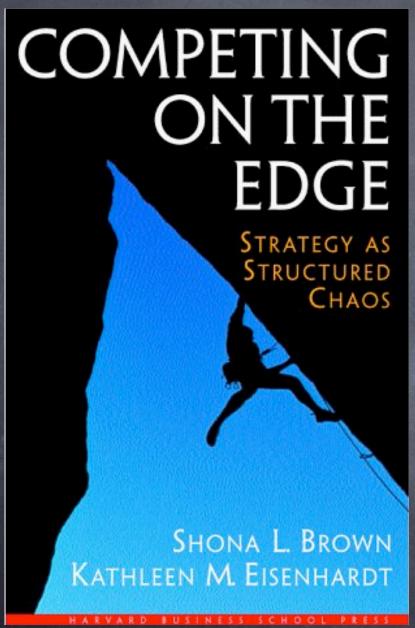


Christopher Alexander

...a process which brings order out of nothing but ourselves; it cannot be attained, but it will happen of its own accord, if we let it.



Agilita' Strategica



...traditional approaches to strategy often collapse in the face of rapidly and unpredictably changing industries ... because they <u>over</u>-emphasize the degree to which it is possible to predict ...

... change is <u>the</u> striking feature of contemporary business ... the key strategic challenge is **managing** that change.

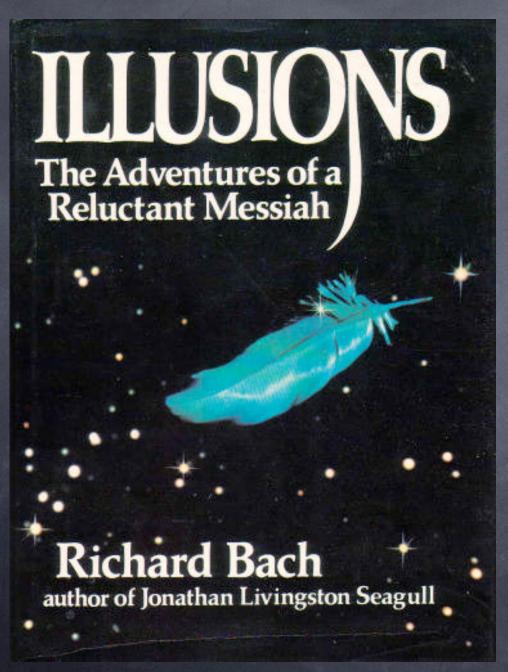


E Allora di Che Parlo?

- o comunico le mie esperienze (a Google e prima) e riflessioni e consigli in merito
 - non aneddoti, ma il "succo"
 - su UN buon modo di fare gestione tecnica dello sviluppo SW
 - certamente ne esistono anche altri, eh!-)
- attacco alcune tesi molto popolari
 - e raccomando alcuni libri che le difendono!-)
- scambio di esperienze, consigli, riflessioni



...e la migliore ragione...:



You teach best what you most need to learn.



Il "Livello" A Cui Parlo







Cosa Ci Motiva

Le idee di Maslow sui bisogni umani sono valide...: ...ma, non vengono necessariamente dal basso verso l'alto!

Transcendenza

Autorealizzazione

Bisogni Estetici

Bisogni Cognitivi

Bisogno di Stima

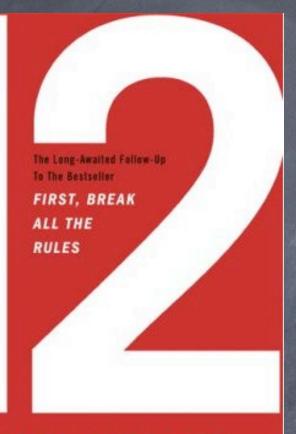
Bisogno di Appartenenza

Bisogno di Sicurezza

Bisogni Fisiologici



Motivazioni sul Lavoro



THE ELEMENTS OF GREAT MANAGING

Based on Gallup's ten million workplace interviews the largest worldwide study of employee engagement

RODD WAGNER & JAMES K. HARTER, PH.D.

- o so cosa ci si aspetta da me
- strumenti, materiali, ecc
- spesso/di solito faccio cio` che so fare meglio
- ho stima, riconoscimenti
- ø il manager si cura di me
- " incoraggia a crescere
- le mie opinioni contano
- o apprezzo la missione
- la qualita è importante



Lo Sviluppo SW = 1 Sgabello

servono tre buone gambe robuste:



giusta intenzione

giusta azione



Lo Sviluppo SW = 1 Sgabello

servono tre buone gambe robuste:



grande



Cosa rende i leader strategici grandi?

- visione profonda e azione incisiva su business model, tecnologie, collaborazioni, personale, agilita`: strategia + esecuzione
- reciproca fiducia, interazione e rispetto con managers e sviluppatori (e i reparti di operazioni, vendita, marketing, ...)
- coraggio, integrita`, umilta`, realismo, orgoglio, ottimismo, prudenza
 - 1. fanno bene il loro (difficile) lavoro
 - 2. mi lasciano fare il mio!



Cosa rende gli sviluppatori eccellenti?

- capacita` di progetto, codifica, testing, debugging; padronanza di algoritmi, linguaggi, strumenti, tecnologie: anzitutto, eccellenza tecnica
- reciproca fiducia, interazione e rispetto col manager e fra loro (e UI, QA, ...)
- coraggio, integrita`, umilta`, realismo, orgoglio, ottimismo, prudenza
 - 1. fanno bene il loro (difficile) lavoro
 - 2. mi permettono di fare il mio!



Come Si Fa Ad Averli?-)

Ogni manager sogna di avere grandi leader strategici ed eccellenti sviluppatori, ma, come si fa a trovarne di cosi`validi?

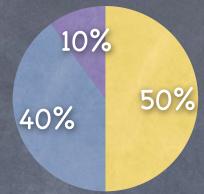
- (1) fortuna (ad es.: se ti capita di lavorare a Google!-)
- (2) scelta (ad es: vieni a lavorare a Google!-)
- (3) coltivali (non mutuamente esclusiva c/1-2!)
 - la fiducia, in particolare, richiede sempre di essere coltivata (e ne riparleremo)
 - puoi sempre insegnare un poco
 - puoi sempre imparare tanto

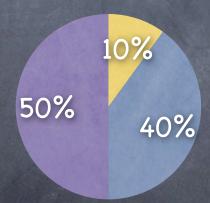


Due Mostri Spaventosi

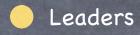


Sviluppare il software sbagliato





Percentuali di colpa



Developers

Manager

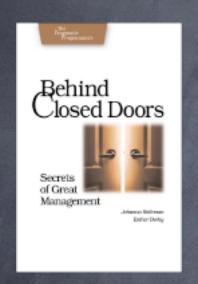
Sviluppare male il software

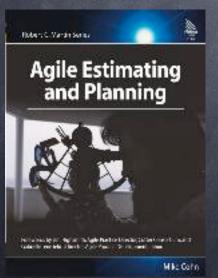


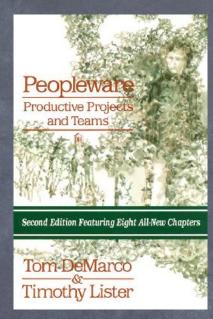


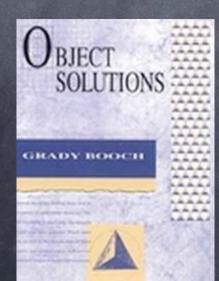
Un modo di imparare...

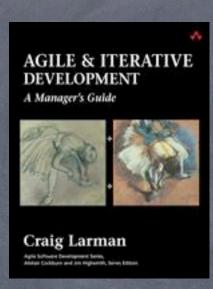
...e` con buoni libri -- ce ne sono tanti!



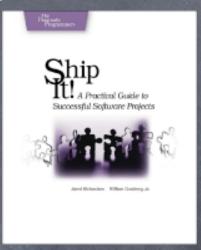


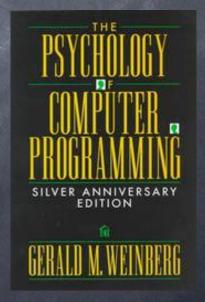








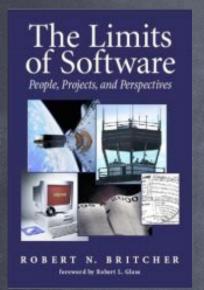


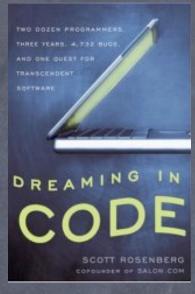


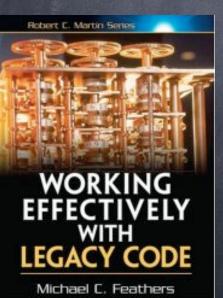


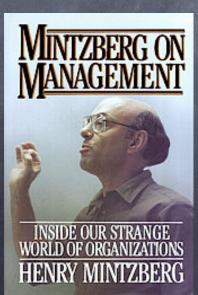
Ars longa, vita brevis...

piu` ottimi libri, che tempo per leggerli!-)

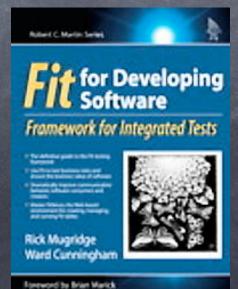


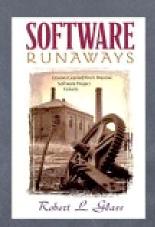


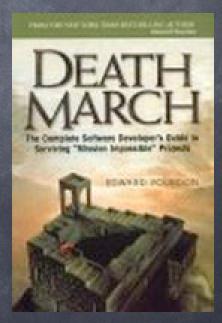














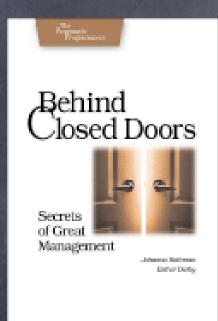
Imparare Dai Libri...

...come tutti sappiamo, richiede il giusto approccio:

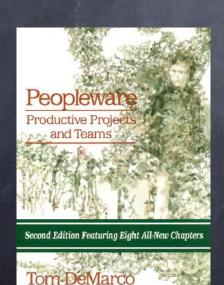
- ø i libri van letti sempre in modo critico
 - aiuta leggerne vari in dissenso fra loro!-)
- o vanno messi alla prova dell'esperienza
 - prova nuove cose e idee un po' alla volta
 - ø il "Big Bang" non solo e` rischioso (!),
 - ma offusca percezione ed analisi
 - ø usa soprattutto la tua esperienza, ma non solo -- usa anche la mia, la sua, la loro...
- NB: imparare da conferenze, presentazioni, o corsi, richiede <u>lo stesso</u> approccio!-)



Un'Opinione Diffusa



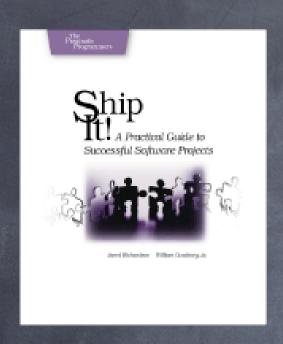
"Once you have four or more people in your group, you can't perform technical work and still be a great manager." (Wk 6)



"Managers are not usually part of the teams that they manage ... leadership just doesn't have much place here." (Ch 23)



E un Dissenso da Essa



"Tech leads split their time between development tasks and management tasks, not working exclusively in either realm." (T.15: Let a tech lead)

A Google, distinguiamo i puri "tech lead" (ingegneri responsabili di un progetto) dagli "uber tech lead" e "tech lead/ manager" ("highly technical managers" == HTM), ma entrambi soddisfano questa definizione.



Un modo di essere un HTM

- supponiamo che un manager M sia tecnicamente un pari degli sviluppatori (progetto, codifica, debugging...)
- M puo` nutrire reciproca fiducia, interazione e rispetto con gli sviluppatori usando se stesso come "risorsa tecnica 'Jolly'"
 - o non per i compiti "divertenti", ma
 - per quelli urgenti che richiedono un altro paio di mani emisferi cerebrali subito,
 - o che siano divertenti o (meglio!-) pallosi
- a 'sto punto di solito arrivano le obiezioni...



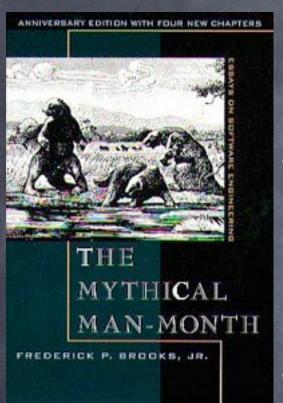
Non Puo' Andare, Perche'...

Se segui criticamente, potresti avere una o piu` di queste obiezioni:

- 1. e la Legge di Brooks?
- 2. ma come si trova il tempo!?
- 3. ma un manager non deve sempre delegare?
- 4. si trascura il "vero" lavoro da manager!
- 5. cosi` si spreca talento tecnico!
- 6. troppe interruzioni, non si puo` mai raggiungere lo "stato di flusso"!
- 7. ...aggiungi le tue personali obiezioni...:-)



1. La Legge di Brooks "Aggiungere programmmatori a



"Aggiungere programmmatori a un progetto software in ritardo lo ritarda ancora di piu"

- Si, ma: tutti scordano sempre di citare anche <u>l'inizio</u> della frase: "Sovrasemplificando in modo oltragioso, asseriamo"...!-)
- poi, basta che <u>non sia</u> in ritardo...!-)

Si basa sul carico ulteriore sugli sviluppatori esistenti per addestrare i nuovi + costo delle comunicazioni in piu`.

Ma: l'HTM e` sempre sostanzialmente al corrente, e comunica sempre, quindi: no overhead re no Brooks' Law



2. Come si trova il tempo?

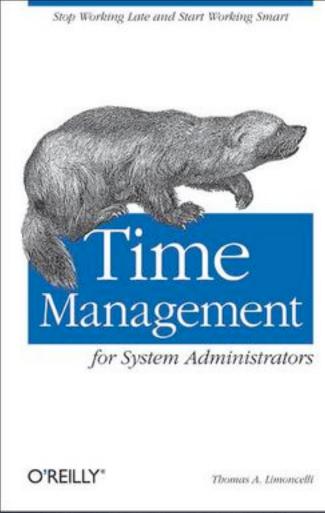
- NON lavorando sempre piu` ore la settimana
 - mira a 40
 - accetta 45
 - 50 e` escluso

(Nota: dico vero tempo di lavoro, al netto di blogging, spuntini, surfing, pranzi...!-)

- onon col telecommuting (faccia-a-faccia e` la forma + efficace di comunicazione, e la comunicazione e` fra gli aspetti piu` cruciali del lavoro di qualsiasi HTM)
- ø il "time management" funziona, se ben fatto



Time Management per ...



- onon e' solo per SA:
 - øalmeno l'80/90% e` utile a sviluppatori e HTM
 - specie se hanno anche compiti operativi
- Limoncelli presenta il nucleo:
 - http://video.google.com/videoplay? docid=7278397109952382318
- idee chiave: focus & interruzioni, lista TODO unica e come gestirla, crearsi abitudini, la prioritizzazione

Aggiungo qualche consiglio specifico per HTM...



Time Mgmt 101 per HTM

- programma molti <u>brevi</u> meeting periodici
 - se un meeting finisce presto, no problem
 - ø si cancella in caso di emergenze
 - ø gestire bene il meeting lo mantiene breve
 - la puntualita` risparmia tempo per <u>tutti</u>
 - mai programmare 2 meeting consecutivi!
 - pensa sempre a chi dovrebbe esserci
 - facile ma errato: "nel dubbio, invita anche quello"!-)
- ø sii sempre pronto a cogliere le opportunita`
 - @ laptop, libri, Blackberry/PDA, QCFPT

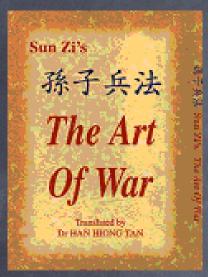


Time Mgmt 102 per HTM

- o considera ogni compito specificamente
 - ø deve davvero essere fatto?
 - sono la persona giusta per farlo?
 - ø quando sarebbe ottimale farlo?
- o non lasciare emergere le emergenze!
 - o un rammendo in tempo salva il calzino
- riserva ~50% del tuo tempo "disponibile" ogni settimana per cose non ancora urgenti
 - o un generale saggio mantiene una riserva
 - o compiti rimandabili in casi di emergenze
 - o non aspettare che diventino urgenti!-)



Metafore Militari





"Pur seguendo il mio proficuo consiglio, avvaliti anche di qualsivoglia circostanza a te favorevole, checche` ne dicano le normali regole." (v. 16)

NB: alcuni generali persero battaglie per non aver <u>mai</u> impegnato le proprie riserve!-)

Von Moltke

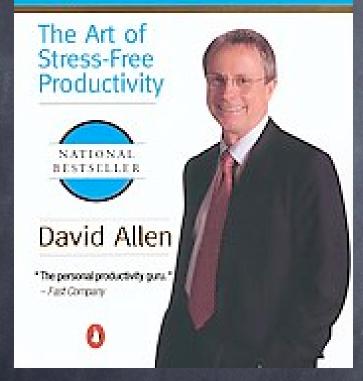
Nessun piano sopravvive al contatto col nemico

Pianificare e` tutto, i piani non sono nulla.



Estremamente Popolare:

GettingThings VDONE



- ødecisamente non-specifico
 (orientato ai manager, ma non
 hi-tech)
- aha molti veri e propri entusiasti, un "movimento"
- http://www.davidco.com/
- oidee chiave: mente come acqua, un singolo "in-basket", flusso strutturato, passi d'azione, "regola dei 2 minuti"

non funziona appieno per me, ma per tanti altri si`!



3. ma un mgr deve delegare

- o certo, ma, delegare cosa?
 - ø delegare non toglie responsibilita`
 - resta <u>sempre</u> aggiornato su tutti i progetti che guidi o coordini!
 - devi fidarti che gli sviluppatori facciano cose giuste -- ma, devi anche far la tua parte, per permettergli di farle!
- una volta che gli sviluppatori vedono che i tuoi contributi tecnici sono eccellenti,
 - <u>e</u> si fidano che gli darai il dovuto credito,
 - oti vorranno coinvolto il + possibile!

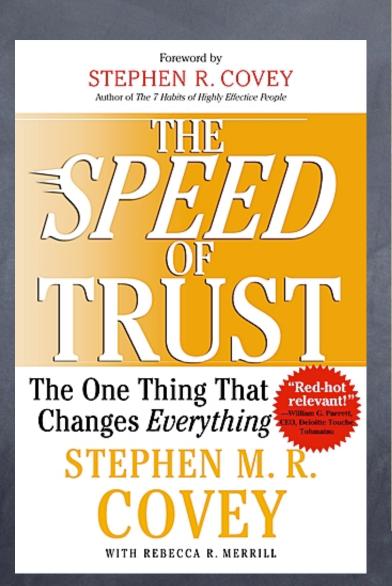
La Fiducia...

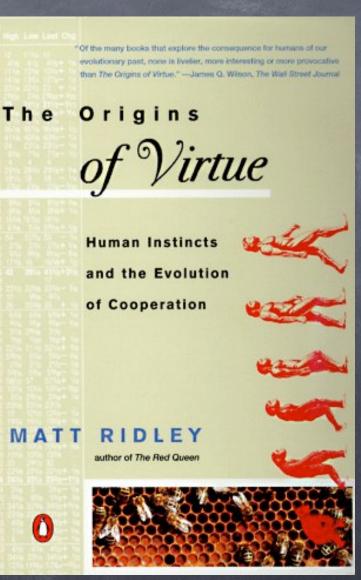
- @ e` reciproca, e si costruisce col tempo
- ø devi guadagnarti la fiducia degli sviluppatori
 - o capacita` tecnica, esercitata regolarmente
 - o interesse vero in loro come individui
 - o dai sempre riconoscimenti e credito
- o loro devono guadagnarsi la tua
 - o capacita` tecnica, integrita`, focus
 - ma: inizia sempre "fidandoti di default"!
 - pre-req: assunzioni MOLTO selettive!-)



Ancora sulla Fiducia

"A fascinating, provocative, and important book." Douglas R. Hofstadter, outhor of Godel, Escher, Bach THE Evolution Cooperation ROBERT AXELROD







4. Si trascura il vero lavoro

- il vero lavoro del manager consiste di questo: coltiva la fiducia, interessati alle persone, aiuta le squadre a formarsi, segui accuratamente tutti i tuoi progetti, aiuta le persone a crescere, concentrati sugli scopi da raggiungere e le loro priorita`
- scrivere unit-test, discutere un progetto, o una terribile sessione di debug, aiutano a svolgere ciascuno di questi compiti!
 - e poi cosi` anche noi ci divertiamo con un poco di hacking al posto giusto, no?-)



Non sono denti d'ingranaggio

- impara tutto quel che puoi sulle specifiche, individuali forze e debolezze delle persone
 - specie i TLs, ma non solo loro
- fa leva sulle loro forze
- ø fa scudo alle debolezze, ma anche:
 - aiutali a crescere, rimuovere debolezze
 - o coaching, corsi, libri, pair progr., ...
 - ø e` una maratona, non i 100 metri piani!
- fare richieste irragionevoli puo causare burnout (attento al burnout!!!), ma...
 - fare solo richieste ragionevoli non offre nessuna sfida (stretch goals)



5. Spreco di talento tecnico?

- o non e` uno spreco, e` un leveraging!
- in certe aziende il management e` per chi non ha piu` nulla da contribuire sul piano tecnico... ma non in quelle di <u>successo!</u>-)
- ma questo vale solo in fase di progetto"
 - per nulla!
 - "il diavolo sta nei dettagli"
 - e dove c'e un diavolo da combattere, la e dove servono i migliori esorcisti!-)

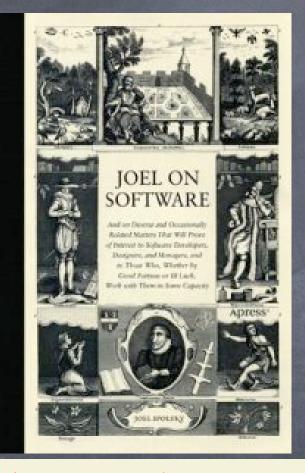


6. Interruzioni e Flusso

- 🛮 v. Limoncelli su come gestire le interruzioni
- ne restano comunque tante da "servire"...:
 - ø tienti fuori dai "percorsi critici" (o garantisci i routing alternativi)
 - o impare a capire cosa puo` aspettare 1 ora
 - impara a "salvare qualcosa sullo stack", dare immediata attenzione a qualcosa d'altro, pop dello stack e continua liscio
- alla fine, e` possibile che uno non sia proprio fatto per multitasking -- e fare il manager ne richiede sempre parecchio!-(



Per consigli vari assortiti



you have to make a schedule... no programmer wants to... most are only doing it because their boss made them do it, halfheartedly, and nobody actually believes the schedule...



Pianificazione e Schedule

- ascolta Joel: la tecnologia appropriata e` una spreadsheet (o whiteboard+post-it, o cartoncini...), non diagrammi PERT/GANTT
 - scegli compiti a grana molto fine (per combattere l'ottimismo degli sviluppatori, e il tuo personale!-)
 - considera nella schedule le vacanze, i corsi, le malattie (combatti attivamente per evitare il burnout!)
- ascolta Cohn: stima la dimensione, deriva la durata (e, usa unita` arbitrarie * velocita`)

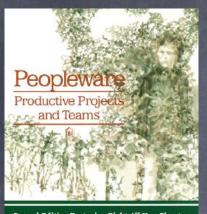


Strumenti appropriati

- o cosa deve essere uniforme nella "squadra"?
 - ø stile del codice, nomi, spazi, idiomi
 - OS, librerie, framework di test, controllo dei sorgenti, issue tracking
 - ma NON necess. editor, debugger, IDE...
- strumento + importante: un buon sistema di controllo dei sorgenti (e script accessori per: continuous build, automatic tests, ...)
- subito dopo: sistema di issue-tracking ben integrato col sistema di controllo sorgenti

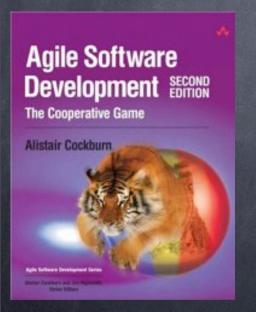


Uffici o Open Space?



Second Edition Featuring Eight All-New Chapters

Tom DeMarco Timothy Lister

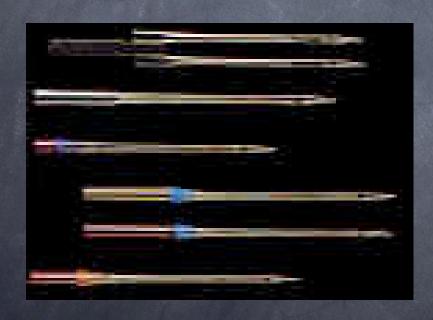


- DeMarco e Lister lottano per uffici monopersona e con porta chiudibile
 - ...allora meglio il telecommuting...
 - ...e la comunicazione?
- Cockburn: osmosi e correnti d'aria
 - ø il problema dello "status"
 - "radical colocation"
 - open-space? una-stanza-persquadra? "caverne e piazza"? "buoni" cubicoli? o che altro?
 - ø e` un problema aperto... o tr.e!-)



E la pallottola d'argento?!

o non c'e`, ma tanti piccoli spilli tengono lontani i mostri!-)







2?

A.

Google

"The Timeless Way of Building", C. Alexander "Competing on the Edge", S. Brown, K. Eisenhardt "The Elements of Great Managing", R. Wagner, J. "Illusions", R. Bach "Behind Closed Doors", J. Rothman, E. Derby "Peopleware", T. DeMarco, T. Lister "Agile & Iterative Development", C. Larman 'Ship It!", J. Richardson, W. Gwaltney "Agile Estimating and Panning", M. Cohn "Object Solutions", G. Booch "Agile Software Development", R. Martin "The Psychology of Computer Programming", G. Weinberg "The Limits of Software", R. Britcher "Dreaming in Code", S. Rosenberg "Project Management", S. Berkun "Software Runaways", R. Glass "Working Effectively with Legacy Code", M. Feathers "Mintzberg on Management", H. Mintzberg "FIT for Developing Software", R. Mugridge, W. Cunningham "Death March", E. Yourdon "The Mythical Man-Month", F. Brooks "Time Management for System Administrators", T. Limoncelli "The Art of War", Sun Zi "How to Lose a Battle", B. Fawcett "Getting Things Done", D. Allen "Trust", F. Fukuyama "The Evolution of Cooperation", R. Axelrod "The Speed of Trust", S. Covey "The Origins of Virtue", M. Ridley "Joel on Software", J. Spolsky "Agile Software Development: The Cooperative Game", A. Cockburn

