



Obsolescence

The part of IT you don't care for

June 2015

Darius Blasband – darius@raincode.com

The company



- Based in Brussels, Belgium
 - Founded in 1998, 33 people
 - Privately owned,
 - Profitable from day one
 - Strong relationship with Microsoft
- RainCode USA incorporated in April 2012
 - To better serve our biggest market (> 55%)
- We do compilers and languages
 - Compilation, migrations and quality control
 - PACBASE, CA-DATACOM and COOL:Gen as major service activities
 - But also code generation for DSLs and model-based development (Mastercard...)
- An evolving business model...
 - From indirect sales through partnerships...
 - .. to an increasing share of direct sales
- End user references:
 - DVV Insurance (Brussels), PWGSCA, Sun Life Financial, GE Aviation (USA), Mastercard (Belgium), Wells Fargo (USA), UBS Warburg (Zurich), CRI Trade (USA), SAIC (USA), ALD Automotive (France), Société Générale (France), San Mateo County (USA), NRI (Japan), Alstom (World wide), Sopra (World wide), Volvo-IT (France), Microsoft (World wide), MAAF (France), Sage group (France), MBank (Poland), PriceWaterhouse Coopers (Netherlands), SDC (Denmark), Elektron (Faroe Islands), mBank (Poland), Pole Emploi (Nantes, France), Natixis (France)...



sage



VOLVO

Scottrade

ALSTOM



Sun 
Life Financial



Sopra
group





Looking for
bright people
for our
technical staff
in Brussels...
and elsewhere

Worth remembering...

Owning a software system costs money

- Maintenance
- Know-how
- Liability

This cost increases with magnitude

- Goes without saying

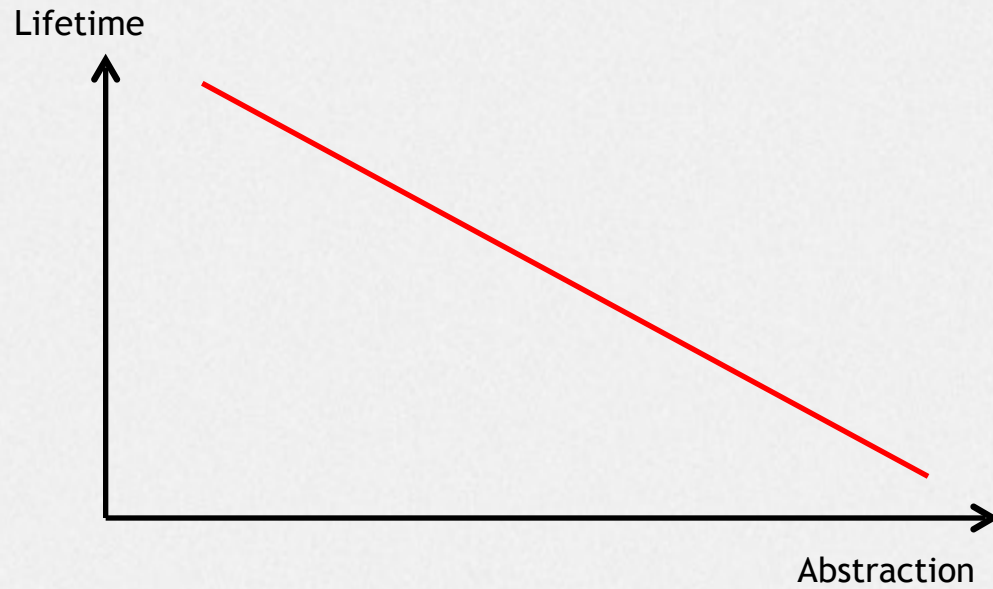
...and with technological dependencies

- Correlated with time
- Not felt before timescales hit the first decade
- Fragility

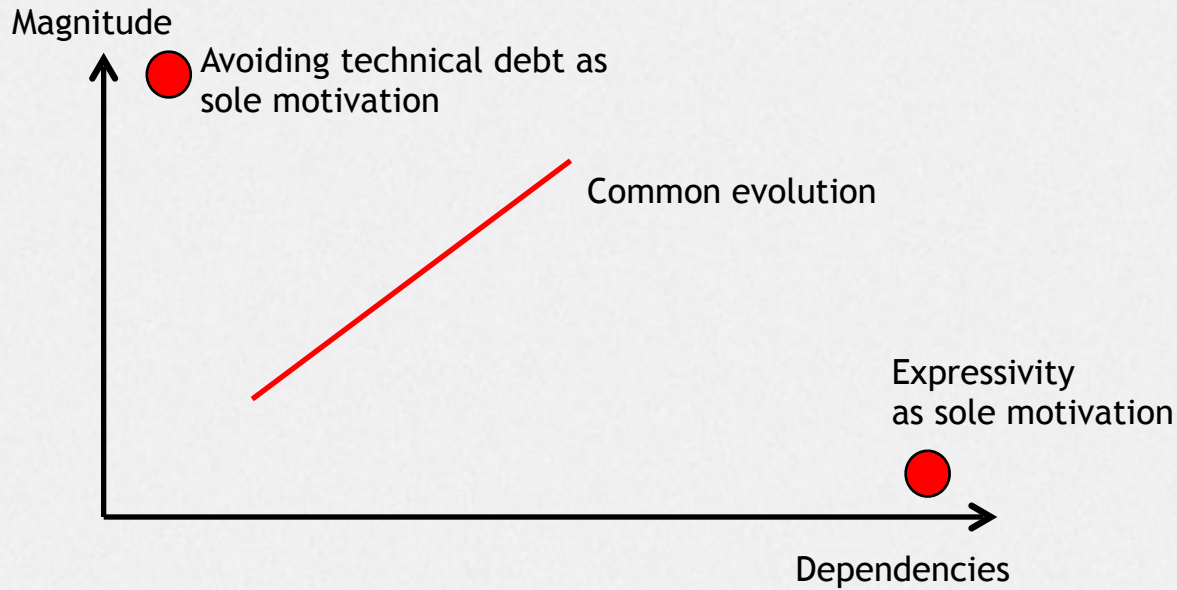
In comparison....

- Development costs are negligible
- Optimizing development productivity is often a misguided strategy
- As it usually implies more numerous and dubious dependencies

An inconvenient negative correlation



The cost of an IT system, along two axis



If you were to develop a system which must be available for 30 years to come, which language(s) would you use?

(You're not gonna like this)

- Will java still be around in 20 years ?
 - Probably
- Can a Java application survive 20 years easily ?
 - Definitely not
- Because it lives in an ecosystems of components
 - Struts, Spring, Ajax, Eclipse (Swt), Jboss, XML parser, Hibernate, etc...
 - Which all evolve independently
 - Kind of a super legacy with distributed responsibilities
- A Java application goes through a series of technical migrations, to update components, deal with conflicts,
 - Fragile
- .NET, for all its flaws, does not suffer from this
 - Or not to the same degree



- Before worrying about obsolescence...
 - What kind of timeframe are we talking about?
- How long does a typical application live?
 - Before falling into oblivion?



- Depending on the domain?
 - Industrial?
 - Banking?
 - Insurance?
- We essentially don't know
 - For banking and insurance the median/average lifetime may very well turn around 100 or 150 years
 - But we'll have to wait before we're sure
 - At least 100 or 150 years 😊
- For all practical purpose:
 - Undetermined in the future

There is such a thing as too much, too old software

- When cost exceeds value

Shows in multiple areas

- Balance sheet
- Head count
- Flexibility

Even less of a platitude

Rewriting a software system...

- Fails more often than not
- Counter intuitive
- But experimentally sound

Replacing it by a package is no picnic either

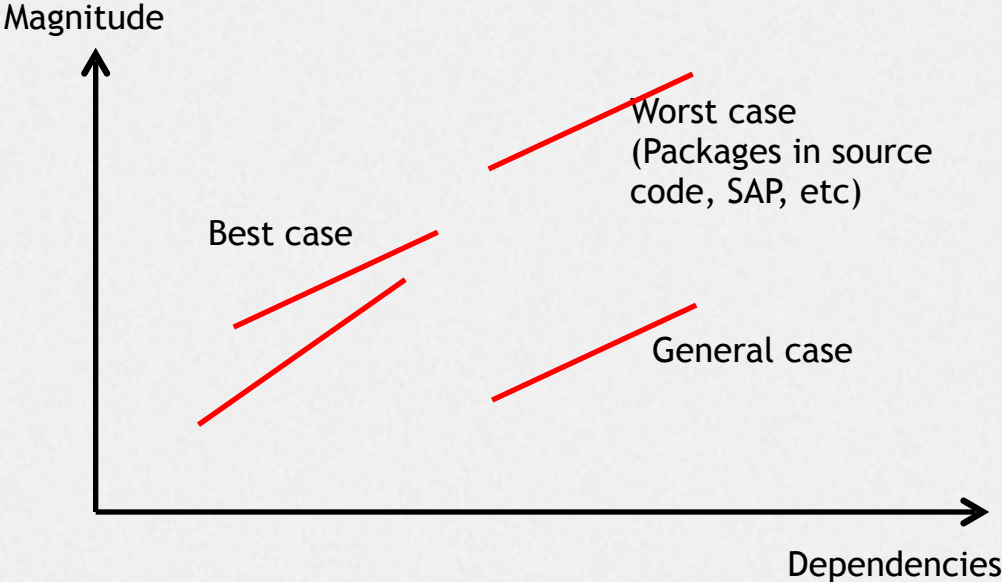
- More on this later...

System decommissioning is a nightmare

- Untapped business opportunity

- Why make such a drama of it?
 - Let's just redevelop everything every 10-15 years
 - Time is no longer an issue!
 - Intuitively reasonable
- But...
- **Factor 1: volume**
 - These systems took 20+ years to develop
 - Redeveloping them in 3 to 5 years is insanely optimistic
 - And already way too long
 - Because productivity has not improved significantly
- **Factor 2: price**
 - Guess what: money is always an issue
- **Factor 3: continuity**
 - The application must keep evolving during redevelopment
 - Moving target
 - Data
 - Integration with external systems
 - Evolving at their own pace
- **All in all:**
 - Redevelopment projects seem to fare even worse than plain developments from scratch

Packages



- **Groupe SNCB:**
 - 400 million euros on a SAP implementation
 - 6.000 users
 - One Mercedes Sedan per user (Leather seats, onboard computer, bar, etc.)
- **40 euros for every Belgian**
 - Babies, infants and crippled
- **And this does not even start contributing to running trains**
 - Just organizational overhead
- **And it is not over yet**
 - Maintenance costs
 - Evolving needs
 - A SAP-fueled legacy
- **Still, presented as a success !**
 - By SAP themselves!
- **Admittely, could have been even worse**
- **But the how rational is it?**
- **Cost vs. value**

- **Poorly mastered issue**
 - Sold by salespeople with quarterly quotas
 - And managed for 20+ years by those naïve enough to listen
- **An extreme form of dependence**
 - Where the organization bends rather than let the software adjust
 - Negating competitive advantage
- **More violence than one may think**
 - No one is too big to fail
 - Nor SAP, IBM, Oracle, Google
 - On a 20 years horizon, anyone can die (Compaq, Wang, Sun, Informix, Digital, Lotus)
- **Not limited to enterprise-level systems**
 - SaaS applications only speed up the process
 - Google's graveyard (Google wave?)
 - Mobile apps turbocharging it
- **This is only the beginning !**
 - Twitter, Facebook, MySpace, Salesforce, Gmail, are uncontrolled liabilities

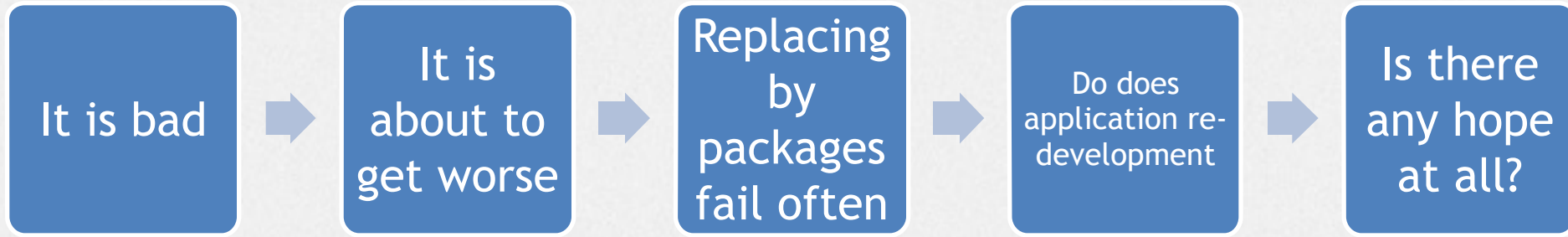
Organizations can die under the burden of their software

- Soaring costs and diminishing value

No such thing as too big to fail

- Extremistan and mediocristan

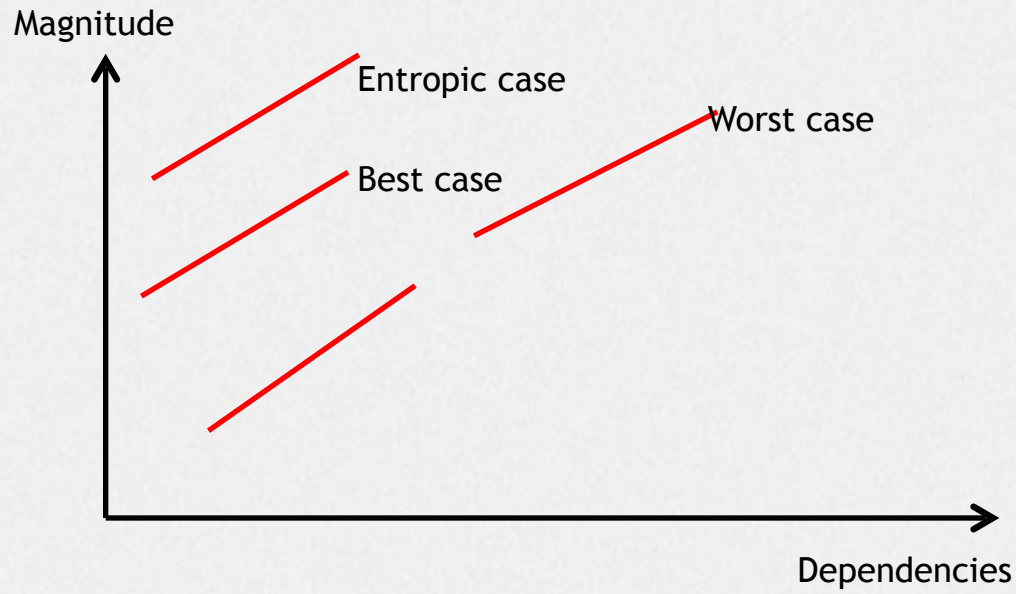
(Nassim Nicholas Taleb)



- Technical migrations
 - In the sense of dealing with technical dependencies
- Better than nothing
- Ungluing applications from ageing infrastructures
 - Programming languages
 - Databases
 - Hardware
 - TP monitor
- Divided in a large number of niche markets
- Two enemies
 - Entropy
 - And dependencies...
 - ...whether one replace a dependency by another...
 - ...or whether one addresses a product dependency but ignores a skill dependency

- Transformational
- COBOL to COBOL
 - Behaviour-preserving refactorings
 - Abstract interpretation (but not on an idealized language !)
 - Clone detection
 - Used as a vector for 4GL migration
- CA-DATACOM
 - Call-level to embedded SQL
 - Also based on abstract interpretation
- Misc
 - Progress to Java
 - Pascal to C
 - Flat files to relational
 - Etc.
- Emulation
- The exact opposite of transformational
 - Keep the system as it is
 - Idiosyncrasies and all
- Compilers
 - COBOL and PL/I
 - The “legacy compiler” concept
 - Soon: ASM370 compiler (!!!)
- Emulators
 - CICS
 - IMS
 - CA-DATACOM
- Reproduce complex behaviours
 - Concurrency and locking
 - Bugs !
- A balancing act

Technical migration



```
public void f90Ao00W_M640 () {
wAo00.fill(" ");
wAo00.setValue(aoFile.getAo00());
udFile.setUd00Undkey(aoFile.getAo00Undkey());
udFile.setUd00Recnum(1);
if ( ! (aoFile.getAo00Rechdr() > 450)) {
udFile.setUd00Rechdr(aoFile.getAo00Rechdr());
} else {
udFile.setUd00Rechdr(450);
}
udFile.insert();
udFile.setUd00Rechdr(udFile.getUd00Rechdr() + 15);
udFile.setUd00Half(wAo00.getW Ao00Half1());
wssBegin.setIk("0");
BaseService fbBS;
fbBS = ServicesFactory.create("com.whoever.fb", this);
fbBS.addCallParameter(17, false);
fbBS.addCallParameter(fbActions.getFieldfbProgram(),
false);
fbBS.addCallParameter(fbActions.getFieldfbWrite(),
false);
fbBS.addCallParameter(fbActions.getFieldNullInfosStatus(),
false);
fbBS.addCallParameter(fbWritePositions4, false);
fbBS.execute();
if (statusArea.getFieldOneUd00Status().
isGreaterThan("02") &&
!statusArea.getFieldOneUd00Status().equals("22") &&
!statusArea.getFieldOneUd00Status().equals("23")) {
w_perform("secud_M720");
}
if (statusArea.getFieldOneUd00Status().equals("22") ||
statusArea.getFieldOneUd00Status().equals("23")) {
wssBegin.setIk("1");
}
udFile.insert();
if (wssBegin.getFieldIk().isFilled('0') &&
aoFile.getAo00Rechdr() > 450) {
udFile.setUd00Undkey(aoFile.getAo00Undkey());
}
```

- The idiom issue must be addressed
- Short of which the migration will be pointless

Things can get so much worse: VistA and MUMPS



- VistA is health-care management system

- Develop the St

- Open so

- A monst

- 160 fu

- 963 hc

- An ent

- Potent US pop

- Technical, public and political success

```
%DTC
%DTC ; SF/XAK - DATE/TIME OPERATIONS ;1/16/92 11:36 AM
; ;19.0;VA FileMan;;Jul 14, 1992
D I 'X1!'X2 S X="" Q
S X=X1 D H S X1=%H,X=X2,X2=%Y+1 D H S X=X1-%H,%Y=%Y+1&X2
K %H,X1,X2 Q
;
C S X=X1 Q:'X D H S %H=%H+X2 D YMD S:$P(X1,".",2) X=X_"_"$P(X1,".",2) K X1,X2 Q
S %=#60/100+(%#3600\60)/100+(%\3600)/100 Q
;
H I X<1410000 S %H=0,%Y=-1 Q
S %Y=$E(X,1,3),%M=$E(X,4,5),%D=$E(X,6,7)
S %T=$E(X_0,9,10)*60+$E(X_0000",11,12)*60+$E(X_00000",13,14)
TOH S %H=%M>2&'(%Y#4)+$P("^31^59^90^120^151^181^212^243^273^304^334","^",%M)+%D
S %='%M!'%D,%Y=%Y-141,%H=%H+(%Y*365)+(Y\4)-(Y>59)+%,%Y=$S(:-1,1:%H+4#7)
K %M,%D,% Q
;
DOW D H S Y=%Y K %H,%Y Q
DW D H S Y=%Y,X=$P("SUN^MON^TUES^WEDNES^THURS^FRI^SATUR","^",Y+1)_"DAY"
S:Y<0 X="" Q
7 S %=%H>21608+%H-.1,%Y=%\365.25+141,%=#365.25\1
S %D=%+306#(%Y#4=0+365)#153#61#31+1,%M=%-D\29+1
S X=%Y_"00"+%M_"00"+%D Q
;
YX D YMD S Y=X_% G DD^%DT
YMD D 7 S %=$P(%H,".",2) D S K %D,%M,%Y Q
```

to a huge

MUMPS

straction

n dollars a

- A total figure of 500 millions is often cited
- To no avail

Obsolescence is like caring for the environment

It is not always fun
But it is here to stay
Get used to it !


Our future will be about dealing with the past


A red pushpin is pinned to the top center of a yellow sticky note.


Thank
You!




RAINCODE HEADQUARTERS


 45 Rue de la Caserne
B-1000 Brussels, Belgium


 +32 2 522 06 63

 info@raincode.com

RAINCODE USA

 13245 Atlantic Boulevard. Suite 4-263
Jacksonville, FL 32225, USA

 +1 412.552.8207

 info@raincode.com

www.raincode.com

