

Desenvolvimento de Projectos com Tecnologias Espaciais

Algumas Reflexões

Joana Simões ¹

¹e-GEO, CASA

May 23, 2013

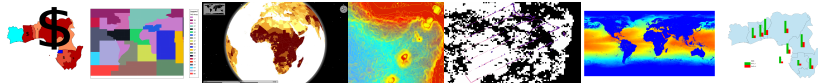


Tabela de Conteúdos

- 1 Introdução
- 2 Escolha de Ferramentas Adequadas
- 3 FOSS
- 4 Informação Espacial
- 5 AGILE
- 6 Sumario
- 7 Referências

Introdução

Quem são os principais destinatários desta comunicação?

Introdução

Quem são os principais destinatários desta comunicação?

- pessoas que desenvolvem projectos de software, com tecnologias espaciais;

Introdução

Quem são os principais destinatários desta comunicação?

- pessoas que desenvolvem projectos de software, com tecnologias espaciais;
- pessoas cuja equipa onde estão integradas não se dedica como actividade primária à implementacao de software (ex: câmara municipal, escola, instituto de investigação aplicada, etc);

Escolha de Ferramentas Adequadas

The right tool, for the right job

Do ponto de vista do *developer*:

Escolha de Ferramentas Adequadas

The right tool, for the right job

Do ponto de vista do *developer*:

- não existem soluções "perfeitas"!

Escolha de Ferramentas Adequadas

The right tool, for the right job

Do ponto de vista do *developer*:

- não existem soluções "perfeitas"!
- avaliar qual é a melhor tecnologia para os objectivos que se pretendem;

Escolha de Ferramentas Adequadas

The right tool, for the right job

Do ponto de vista do *developer*:

- não existem soluções "perfeitas"!
- avaliar qual é a melhor tecnologia para os objectivos que se pretendem;
- ter em conta a linguagem/ferramenta em que se esta mais "confortável", e pesar o tempo/esforço em aprender uma nova;

Escolha de Ferramentas Adequadas

The right tool, for the right job

Do ponto de vista do *developer*:

- não existem soluções "perfeitas"!
- avaliar qual é a melhor tecnologia para os objectivos que se pretendem;
- ter em conta a linguagem/ferramenta em que se esta mais "confortável", e pesar o tempo/esforço em aprender uma nova;
- remover os custos ligados à aquisição de software proprietário;

Escolha de Ferramentas Adequadas (+)

Do ponto de vista dos recipientes:

Escolha de Ferramentas Adequadas (+)

Do ponto de vista dos recipientes:

- normalmente os sistemas mais "simples" são aqueles que se aguentam mais tempo; sistemas muito sofisticados têm invariavelmente uma manutenção complicada e cara!

Escolha de Ferramentas Adequadas (+)

Do ponto de vista dos recipientes:

- normalmente os sistemas mais "simples" são aqueles que se aguentam mais tempo; sistemas muito sofisticados têm invariavelmente uma manutenção complicada e cara!
- evitar tecnologias *cutting edge*, que ainda não possuem uma base de conhecimento sólida em todo o mundo;

Escolha de Ferramentas Adequadas (+)

Do ponto de vista dos recipientes:

- normalmente os sistemas mais "simples" são aqueles que se aguentam mais tempo; sistemas muito sofisticados têm invariavelmente uma manutenção complicada e cara!
- evitar tecnologias *cutting edge*, que ainda não possuem uma base de conhecimento sólida em todo o mundo;
- sempre que possível, optar por standards (ex: OGC) e "fugir" de formatos e software proprietário; os standards em princípio garantem uma continuidade no tempo e no espaço;

Escolha de Ferramentas Adequadas (+)

Do ponto de vista dos recipientes:

- normalmente os sistemas mais "simples" são aqueles que se aguentam mais tempo; sistemas muito sofisticados têm invariavelmente uma manutenção complicada e cara!
- evitar tecnologias *cutting edge*, que ainda não possuem uma base de conhecimento sólida em todo o mundo;
- sempre que possível, optar por standards (ex: OGC) e "fugir" de formatos e software proprietário; os standards em princípio garantem uma continuidade no tempo e no espaço;
- pensar também nas restrições de *hardware* dos recipientes: têm acesso à internet? têm multi core, que resolução têm?

Software Livre e de Código Aberto



<http://www.gnu.org/philosophy/free-sw.html>

Para além do custo "zero" e das motivações "éticas":

Software Livre e de Código Aberto



<http://www.gnu.org/philosophy/free-sw.html>

Para além do custo "zero" e das motivações "éticas":

- **custos de *hardware* mais baixos:** geralmente o software FOSS requer menos capacidade computacional para realizar as mesmas tarefas que em servidores "convencionais" ou *workstations*.

Software Livre e de Código Aberto



<http://www.gnu.org/philosophy/free-sw.html>

Para além do custo "zero" e das motivações "éticas":

- **custos de *hardware* mais baixos:** geralmente o software FOSS requer menos capacidade computacional para realizar as mesmas tarefas que em servidores "convencionais" ou *workstations*.
- **gestão de licenças simplificada:** podem obter-se quantas licenças se quiser, para instalar em toda a parte, o que quer dizer que a produtividade não é afectada por questões de licenças.

Software Livre e de Código Aberto (+)

- **amplo suporte:** o suporte para FOSS é muitas vezes superior ao de soluções proprietárias. Isto deve-se a que existem dois níveis de suporte: o gratuito, providenciado pela comunidade online (em crescimento) e o pago, que muitas companhias agora disponibilizam (ex: Novell).

Software Livre e de Código Aberto (+)

- **amplo suporte:** o suporte para FOSS é muitas vezes superior ao de soluções proprietárias. Isto deve-se a que existem dois níveis de suporte: o gratuito, providenciado pela comunidade online (em crescimento) e o pago, que muitas companhias agora disponibilizam (ex: Novell).
- **qualidade de software:** o processo de revisão pelos pares e os standards da comunidade, adicionados ao facto de que o código é revelado a todos, tendem a promover a "excelência" em *design* e *coding*;

Software Livre e de Código Aberto (+)

- **amplo suporte:** o suporte para FOSS é muitas vezes superior ao de soluções proprietárias. Isto deve-se ao facto de que existem dois níveis de suporte: o gratuito, providenciado pela comunidade online (em crescimento) e o pago, que muitas companhias agora disponibilizam (ex: Novell).
- **qualidade de software:** o processo de revisão pelos pares e os standards da comunidade, adicionados ao facto de que o código é revelado a todos, tendem a promover a "excelência" em *design* e *coding*;
- **"vida" extendida:** a disponibilidade do código fonte e o direito de o modificar ("liberdade" numero 1) possibilita o melhoramento ilimitado do software. Também possibilita portá-lo para um novo *hardware* ou sistema operativo. O direito de distribuir as versões modificadas ("liberdade" 3) possibilita as actualizações frequentes.

Software Livre e de Código Aberto (+)

Milestone na historia do FOSS em Portugal:

Software Livre e de Código Aberto (+)

Milestone na historia do FOSS em Portugal:

Recentemente o Tribunal anulou um concurso público relativo ao licenciamento e manutenção de software Microsoft, lançado por uma câmara municipal.



Ler a história completa: <http://tinyurl.com/bx42shh>

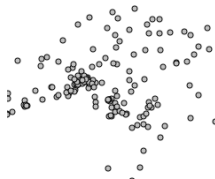
Identificar a Informação Espacial

A informação espacial pode já estar incluída nos dados, embora os *stakeholders* não estejam cientes disso. Há que identificá-la e representá-la de forma adequada.



Exemplo: Pontos como Coordenadas

	PK_UID	LAT	LONG	Geometry
1	1	7.183300	1.983300	BLOB sz=60 GEOMETRY
2	2	7.200000	2.066700	BLOB sz=60 GEOMETRY
3	3	6.350000	2.433300	BLOB sz=60 GEOMETRY
4	4	9.705000	1.666900	BLOB sz=60 GEOMETRY
5	5	11.134200	2.938600	BLOB sz=60 GEOMETRY
6	6	9.350000	2.616700	BLOB sz=60 GEOMETRY
7	7	6.483300	2.616700	BLOB sz=60 GEOMETRY
8	8	5.466700	10.416700	BLOB sz=60 GEOMETRY
9	9	5.933300	10.166700	BLOB sz=60 GEOMETRY
10	10	4.583300	13.683300	BLOB sz=60 GEOMETRY
11	11	5.450000	10.066700	BLOB sz=60 GEOMETRY
12	12	4.050300	9.700000	BLOB sz=60 GEOMETRY
13	13	2.900000	11.150000	BLOB sz=60 GEOMETRY
14	14	3.800000	10.133300	BLOB sz=60 GEOMETRY

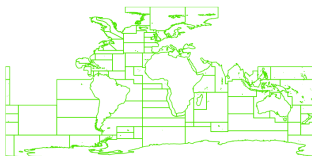


Exemplo: Pontos como Coordenadas (+)

WKT: POINT(1.9833 7.1833)
EWKT: SRID = 4326; POINT(1.98329997067.1833000183)
SVG: x="1.9833" y="-7.1833"
KML: < Point > < coordinates > 1.9832999706, 7.1833000183 < /coordinates > < /Point >
GeoJSON: "type" : "Point", "coordinates" : [1.9832999706, 7.1833000183]

Exemplo: Polígonos como Atributos Nominais

INVISIGOTH/SQ_Fishing_Zones				
ID	Description	Name	NameENG	OLD_CODE
1	missing	Fishing Zone 1	Fishing Zone 1	1
2	missing	Fishing Zone 2	Fishing Zone 2	2
3	missing	Fishing Zone 3	Fishing Zone 3	3
4	missing	Fishing Zone 4	Fishing Zone 4	4
5	missing	Fishing Zone 5	Fishing Zone 5	5
6	missing	Other	Other	998
7	missing	Unknown	Unknown	999
8	missing	missing	missing	missing
9	missing	n/a	missing	missing
*	NULL	NULL	NULL	NULL



http://data.fao.org/maps/ows?SERVICE=WMS&REQUEST=GetMap&layers=GEONETWORK:fa_subareas_31627&width=800&height=600&FORMAT=image/png&CRS=EPSG:4326&BBOX=-180,-90,180,90

Representar a Informação Espacial

Em última instância a forma como representamos os dados espaciais, vai determinar as operações que podemos fazer com eles.

Representar as Informação Espacial (+)

Uma "estação" de arrasto (recolha de amostras de pesca), pode ser armazenada como:



Representar as Informação Espacial (+)

Uma "estação" de arrasto (recolha de amostras de pesca), pode ser armazenada como:

- uma sequência de pontos;



Representar as Informação Espacial (+)

Uma "estação" de arrasto (recolha de amostras de pesca), pode ser armazenada como:

- uma sequência de pontos;
- uma linha;



Representar as Informação Espacial (+)

Uma "estação" de arrasto (recolha de amostras de pesca), pode ser armazenada como:

- uma sequência de pontos;
- uma linha;
- um polígono.



Representar a Informação Espacial (+)

Se queremos calcular a área arrastada, qual é a representação mais adequada?

Representar a Informação Espacial (+)

Se queremos calcular a área arrastada, qual é a representação mais adequada?

```
pts: select area( buffer(Makeline(geometry), 0.5) ) from trawl_points a order by rowid
```

Representar a Informação Espacial (+)

Se queremos calcular a área arrastada, qual é a representação mais adequada?

```
pts: select area( buffer(Makeline(geometry), 0.5) ) from trawl_points a order by rowid
```

```
linha: select area( buffer(a.geometry, 0.5) ) from trawl_lines as a
```

Representar a Informação Espacial (+)

Se queremos calcular a área arrastada, qual é a representação mais adequada?

```
pts: select area( buffer(Makeline(geometry), 0.5) ) from trawl_points a order by rowid
```

```
linha: select area( buffer(a.geometry, 0.5) ) from trawl_lines as a
```

```
poligono select area(geometry) from trawl_polygons
```

AGILE

Agile é um conjunto de práticas de desenvolvimento de software, de forma **iterativa** e **incremental**, resumido nos 12 princípios do manifesto (2001):

<http://www.agilemanifesto.org/principles.html>

AGILE

Agile é um conjunto de práticas de desenvolvimento de software, de forma **iterativa** e **incremental**, resumido nos 12 princípios do manifesto (2001):

<http://www.agilemanifesto.org/principles.html>

- favorecer indivíduos e interações, em detrimento de processos e ferramentas;

AGILE

Agile é um conjunto de práticas de desenvolvimento de software, de forma **iterativa** e **incremental**, resumido nos 12 princípios do manifesto (2001):

<http://www.agilemanifesto.org/principles.html>

- favorecer indivíduos e interações, em detrimento de processos e ferramentas;
- favorecer software que funciona, em detrimento de extensa documentação;

AGILE

Agile é um conjunto de práticas de desenvolvimento de software, de forma **iterativa** e **incremental**, resumido nos 12 princípios do manifesto (2001):

<http://www.agilemanifesto.org/principles.html>

- favorecer indivíduos e interações, em detrimento de processos e ferramentas;
- favorecer software que funciona, em detrimento de extensa documentação;
- favorecer colaboração dos *stakeholders*, em detrimento de negociação de contratos;

AGILE

Agile é um conjunto de práticas de desenvolvimento de software, de forma **iterativa** e **incremental**, resumido nos 12 princípios do manifesto (2001):

<http://www.agilemanifesto.org/principles.html>

- favorecer indivíduos e interações, em detrimento de processos e ferramentas;
- favorecer software que funciona, em detrimento de extensa documentação;
- favorecer colaboração dos *stakeholders*, em detrimento de negociação de contratos;
- favorecer resposta à mudança, em detrimento de "seguir um plano";

AGILE (+)

O desenvolvimento AGILE está francamente estabelecido na indústria de software, mas não suficientemente na área dos SIG.



Numa survey realizada em 2008, apenas 23% dos *developers* de SIG utilizavam práticas AGILE (contra 69% dos *developers* de software *mainstream*).

survey: <http://edgehopper.com/results-of-agile-gis-survey/>

Mitos

Alguns "mitos" sobre a implementação de metodologias Agile:

Mitos

Alguns "mitos" sobre a implementação de metodologias Agile:

- só se aplica em equipas de desenvolvimento de software com várias pessoas, ou em projectos de grande dimensão;

Mitos

Alguns "mitos" sobre a implementação de metodologias Agile:

- só se aplica em equipas de desenvolvimento de software com várias pessoas, ou em projectos de grande dimensão;
- é obrigatório implementar todas as práticas Agile;

Mitos

Alguns "mitos" sobre a implementação de metodologias Agile:

- só se aplica em equipas de desenvolvimento de software com várias pessoas, ou em projectos de grande dimensão;
- é obrigatório implementar todas as práticas Agile;
- toma muito tempo;

Ferramentas Agile

Algumas Ferramentas Agile:

- comunicação simples, falada ou escrita (VoIP, email);
- *web-based collaborative editors* (wiki, etherpad);
- *versioning systems* (git, Subversion, etc);
- ferramentas integradas de gestão de projectos (Redmine, Trac, etc);

Ferramentas Agile (+)

Esta é a "melhor" ferramenta Agile:



Roadmap para o "sucesso":

(Alguns) aspectos a ter em conta:



Referências

- Shore, J. "The Art of Agile Development". O'Reilly Media; 1 edition (November 2, 2007)
- Simoes, J. "Some Thoughts on Writing a Scientific Application". CVU, Vol. 4, Issue 2 (May, 2012). url: <http://accu.org/var/uploads/journals/cvu242.pdf>
- Stallman, R. "Free Software, Free Society". FSF (2002). url: <http://www.gnu.org/doc/fsfs-ii-2.pdf>
- <http://www.casa.ucl.ac.uk/joanamargarida/>
- <http://www.doublebyte.net>



Thank You for Listening!