iText and Testing

Feature rich open source

developer’s library for

PDF generation and manipulation

In web and other applications

- Amedee Van Gasse – iText QA

  @amedee

- Michaël Demey – iText Developer

  @mymilkedeek

Introduction

PDF
PDF
Acro form
XFA
Sign
PDF
page 1

iText Core

Create PDF files

Adapt PDF files

Fill in forms

Sign forms
and PDF files

Stamp PDF files

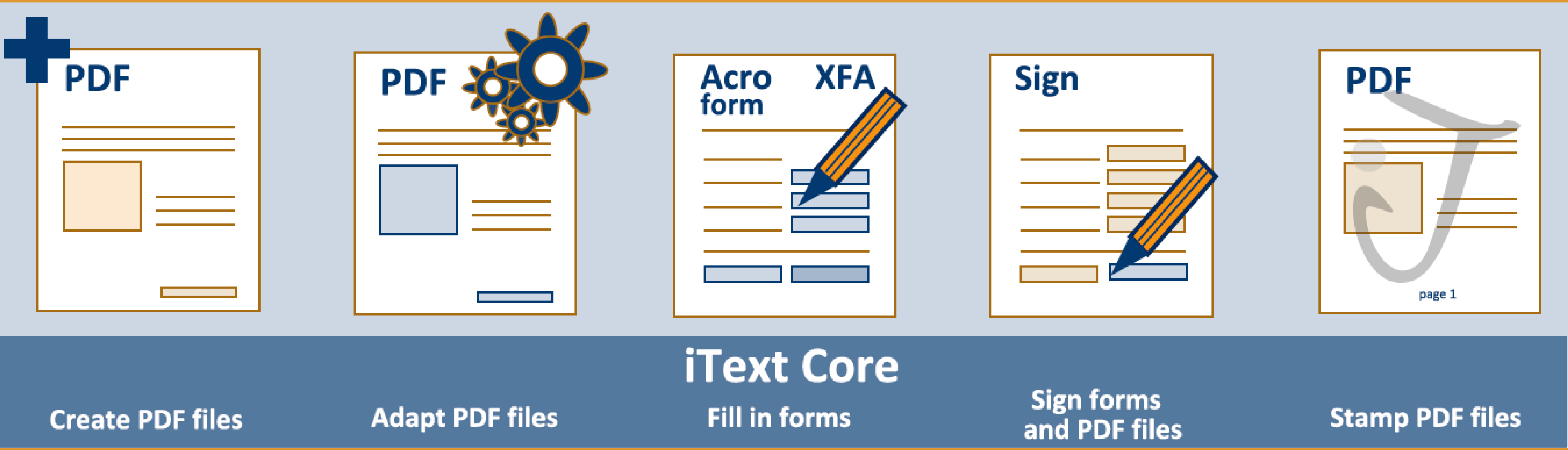
PDF
XML
<XFA> form
Flattening
PDF
Data
Acro form
Flattening
CSS
XHTML
PDF
%PDF
%âãÏÓ

XFA Worker
Dynamic form flattener

XML Worker
Converts (X)HTML and CSS to PDF

iText RUPS
Looks at the structure of your PDF

iText Technology



iText Core

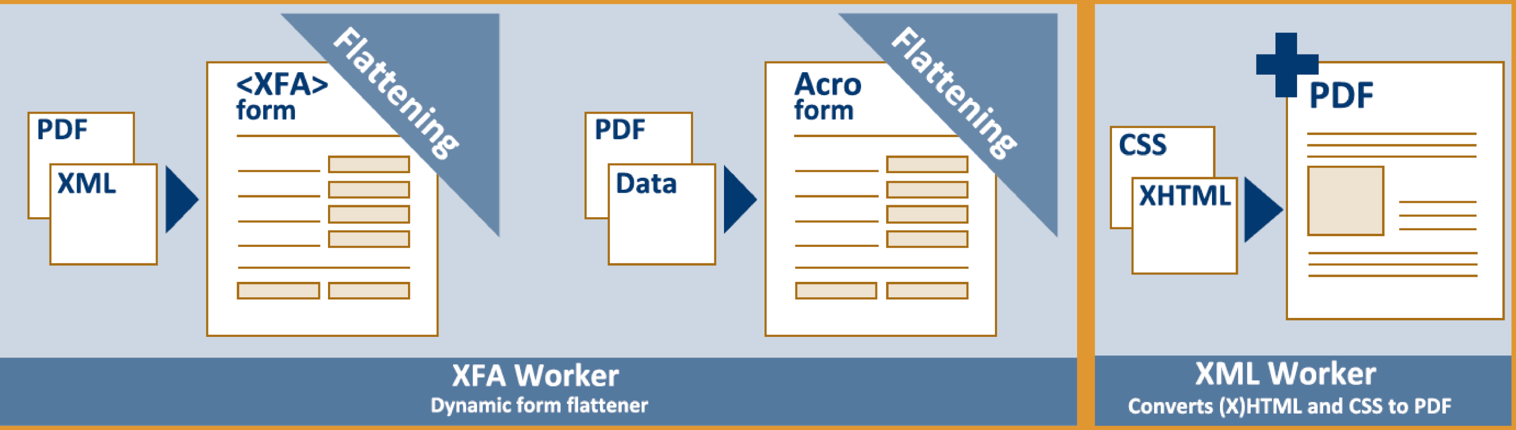
Create PDF files

Adapt PDF files

Fill in forms

Sign forms
and PDF files

Stamp PDF files



XFA Worker
Dynamic form flattener

XML Worker
Converts (X)HTML and CSS to PDF

iText Technology – .NET Port

PDF
PDF
Acro form
XFA
Sign
PDF
page 1

iText Core

Create PDF files

Adapt PDF files

Fill in forms

Sign forms
and PDF files

Stamp PDF files

iText Technology – Android/GAE Port

Early days

- Close to no testing
  - Book samples were the tests
    - Not exhaustive
    - Some samples are outdated (iText in Action 2, released +-4 years ago)
- Checks done manually
  - Visual
  - Structural → RUPS

Early days – Releases

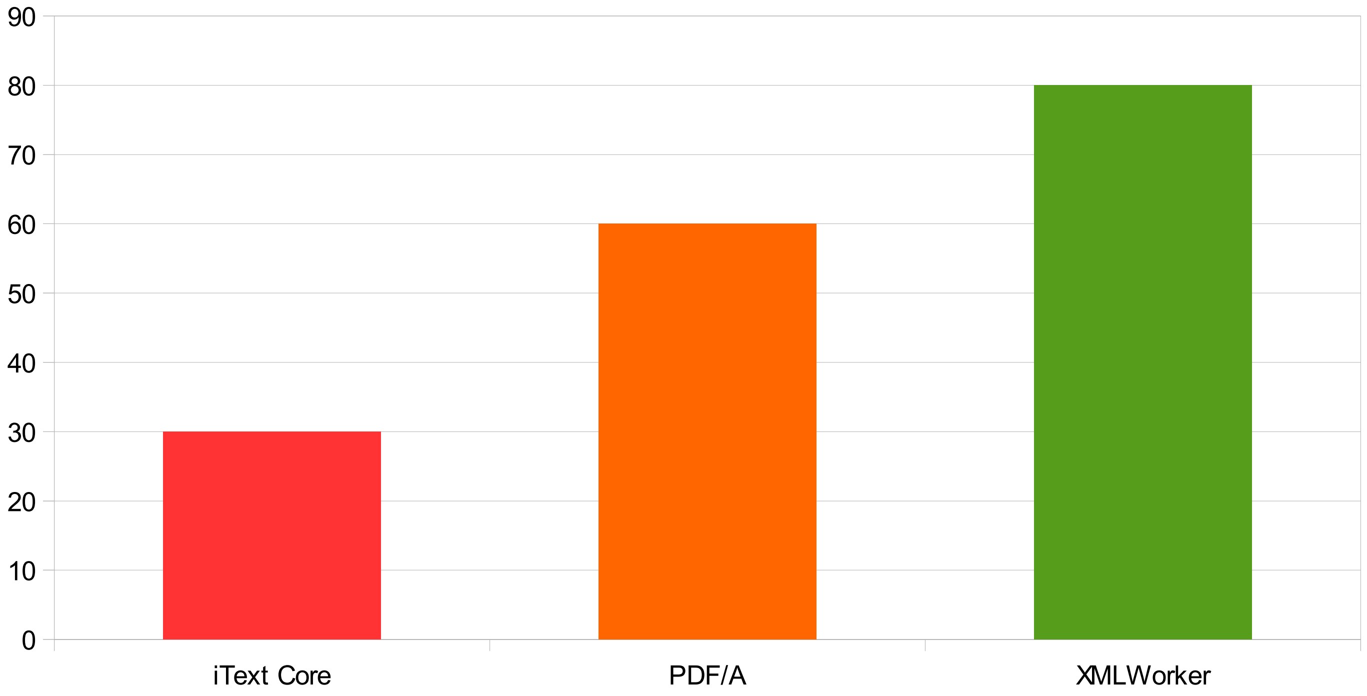
- Week long “event” at iText
- .NET port could take up to two weeks
  - Porting happened after preparing Java release

→ long feedback loop

Current Situation

- 1200+ tests
  - Better, but room for improvement
  - Tools
    - CI: Jenkins and TeamCity
    - Visual Comparisons: GhostScript and ImageMagick

- Test coverage



Current Situation

- New samples (http://itextpdf.com/sandbox)
  - Mainly from questions on StackOverflow, the mailinglist or support
  - @WrapToTest
    - Adds the sample to the test suite

Current Situation

- Performance was slow
  - 40 minute feedback loop from CI tools
    - Currently down to 12 mins
- Reporting is non-existent
- No long term test plan
- Current unit tests (Junit) are actually integration tests
- Mostly black box testing

Current Situation – Issues

- Samples can't be tested separately
- Could be more automated
  - XmlWorker
  - Putting samples on the website is a manual process

Current Situation – Issues

- Reduced to one day “events”
- .NET porting is done in parallel to Java development
  - Quicker feedback from the .NET environment
  - Faster iTextSharp release
- Android/GAE porting
  - Scripted

Current Situation – Releases

- Long term testing plan
  - QA Engineer
- More automation
  - Samples
  - Tests
  - Releases
- Reporting and statistics

Future

- Performance
  - Already improved a lot, but still room for improvement
    - Basic structural checks already implemented
- Code Style standards

Future

Future – Releases

- More automation where possible
- Reduce dependency on SourceForge for code hosting and downloads
  - Github
  - Artifactory

- A lot of work has happened, but a lot of work remains
- Next steps are
  - Set up long term, formal testing plan (following IEEE829 standard)
  - Statstics of current situation (SonarQube)
  - Static code analysis = proactive issue detection (SonarQube, PMD)
  - Unit testing and code coverage (JaCoCo/Cobertura)

Conclusion

Questions?

For Europe,Middle-East,

Asia and Africa

Kerkstraat 108
9050Gentbrugge
BELGIUM

T: +32 92 98 02 31
F: +32 92 70 33 75
E:sales.isb@itextpdf.com

For the US, Canada,

Latin-America and Oceania

1Broadway, 14th floor
Cambridge, MA 02142
USA

T: +1 617 982 2646
F: +1 617 982 2647
E:sales.isc@itextpdf.com

www.itextpdf.com

Contact us!