Virtual User Conference Q3’18

Andy Tryba
CEO, FogBugz
Founder & CEO, Crossover
Founder & CEO, RideAustin
CEO, DNN Corp
CEO, Engine Yard
CEO, Kayako
CEO, Bizness Apps
CEO, My Alerts
Founder & CEO, Think3

Rahul Subramaniam
CEO, DevFactory
Joel, Anil & the Fog Creek Team!
Agenda

About the Acquisition & what does it mean for me?

- Why did we acquire FogBugz?
- Future of FogBugz & DevFactory Products
- Q&A and Summary
Why Fog Creek sold Manuscript (FogBugz)

“We love the vision that DevFactory has established to help developers spend even more time on creative tasks. And most importantly, the Manuscript community will benefit from DevFactory’s focus on customer success and increased investment in modernization of the product.”

- Anil Dash
What does it mean for me?

Massive Investment in the Product

Access to Pre-release DevFactory Products
- DevSpaces
- CodeRead
- CodeCleanup

Relentless Focus on Customer Success
- 20% New Customers
- 80% Existing Customers
Agenda

About the Acquisition & what does it mean for me?

Why did we acquire FogBugz?

Future of FogBugz & DevFactory Products

Q&A and Summary
We built the factory so you don’t have to.

DevFactory is focused on building great products and services for developers.

A number of tasks in the SDLC are non-creative and repeatable. It makes sense to run those operations centrally as AI-assisted assembly lines.
Why we acquired FogBugz

1. We love the customer base
2. We believe in Joel’s original FogBugz vision & want to invest in the product
3. There is huge market opportunity since JIRA sold out
What do mean by ‘JIRA sold-out’?
Abandoned the core: developers in small teams

(and instead focused on big company management overhead & chasing after non-developers adding complexity)
Abandoned the core
Focus away from developers to big company management... Massive complexity

BUSINESS DEALS

Atlassian Buys OpsGenie to Expand in ServiceNow's Market

Atlassian is paying $295 million to acquire OpsGenie, which alerts engineers to software and website outages, Bloomberg reported. The move gets Atlassian, known for its software-development coordination tools, into the market dominated by ServiceNow. Atlassian sees a big opportunity in the IT operations software market, expected to reach $38 billion by 2022, according to Gartner.

Bloomberg | Sep 04, 2018

Atlassian acquires Trello for $425M

4 Features Fogbugz will never implement

- Strangling/Complex workflows
- Inhibiting Permission schemes
- Combined complexity of types, custom fields, Screens, workflows and permissions
- JQL - If you have to create a custom language to search through your issues, it's already too complex.

The minimum set of considerations while creating a workflow in Jira

- Status - where the issue is (for example, "In Progress" or "Under Review")
- Resolution - why the issue is no longer in flight (for example, because it's completed)
- Conditions - control who can action a transition
- Validators - only allow transitions to occur when specific information is provided
- Post Functions - make additional changes to issues, alongside transitions (for example, removing a resolution when an issue is reopened)
- Triggers - automatically activating transitions when specific events take place. For example, moving an issue from "In Progress" to "Under Review" when code is submitted for review
- Workflow properties - setting certain properties for transitions. For example, only displaying resolutions that are relevant to the specific issue type
- Workflow schemes - determining the associations between a workflow and issue type
Atlassian Ticker: TEAM

Moving ‘up-market’: enterprise sell-up playbook
Moving ‘up-market’ with enterprise sell-up playbook

JIRA Price Increases over time

<table>
<thead>
<tr>
<th>Year</th>
<th>Price Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>2002</td>
<td>20%</td>
</tr>
<tr>
<td>2003</td>
<td>40%</td>
</tr>
<tr>
<td>2004</td>
<td>60%</td>
</tr>
<tr>
<td>2005</td>
<td>80%</td>
</tr>
<tr>
<td>2006</td>
<td>100%</td>
</tr>
<tr>
<td>2007</td>
<td>120%</td>
</tr>
<tr>
<td>2008</td>
<td>140%</td>
</tr>
<tr>
<td>2009</td>
<td>20%</td>
</tr>
<tr>
<td>2010</td>
<td>40%</td>
</tr>
<tr>
<td>2011</td>
<td>60%</td>
</tr>
<tr>
<td>2012</td>
<td>80%</td>
</tr>
<tr>
<td>2013</td>
<td>100%</td>
</tr>
<tr>
<td>2014</td>
<td>120%</td>
</tr>
<tr>
<td>2015</td>
<td>140%</td>
</tr>
<tr>
<td>2016</td>
<td>20%</td>
</tr>
<tr>
<td>2017</td>
<td>40%</td>
</tr>
<tr>
<td>2018</td>
<td>60%</td>
</tr>
<tr>
<td>IPO</td>
<td>140%</td>
</tr>
</tbody>
</table>

JIRA continues to add new ‘high-end’ price points

<table>
<thead>
<tr>
<th>Users</th>
<th>Commercial Annual Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>More than 50,000 users</td>
<td>US $450,000</td>
</tr>
</tbody>
</table>

Both Founders made close to $3B
Leaving an enormous market opportunity...

FogBugz

JIRA
($700M/yr in Profit)
Our strategy: Stay true to Joel’s original vision...

Designed for developers – not for management bureaucracy
Designed for simplicity – not for soul-crushing complexity
Designed for speed – not 75 fields just for a simple bug submission
Designed for 2 pizza teams – not expensive dedicated admins

And to start – we’re bringing back the name
Agenda

About the Acquisition & what does it mean for me?

Why did we acquire FogBugz?

Future of FogBugz & DevFactory Products

Q&A and Summary
We’ve heard from customers that they feel the FogBugz product has been neglected...
We are going to change that and will be investing heavily to modernize it.

But it won’t be easy.
In full transparency...

FogBugz has amazing IP built in – but the product is 17 years old...
‘Clone Spec’ details (initial thoughts)

1. Pull FogBugz out of Joel’s basement into AWS
2. Dockerize platform
3. Reduce the LOC from 1.8M by 80%
4. Replace ‘homegrown’ components with best-in-class industry standards:
   - SMTP handler - SES/SendGrid
   - Finance - NetSuite
5. Standardization across cloud & onprem code base
Please have patience with us – this will be hard & time consuming.

There will be problems & mistakes made (we apologize in advance).

But we believe the transition will be worth it. We will perform this in ‘full & open view’ – you get to watch us step by step (and screw-up by screw-up)
Investing & modernize the product matters... But so does ensuring you have the support you need when you need it...
Our operating focus

The Typical Model

1. GROWTH

2. PROFIT

3. CUSTOMER SUCCESS
Our operating focus

The Typical Model

1. GROWTH
2. PROFIT
3. CUSTOMER SUCCESS

The DevFactory Model

1. CUSTOMER SUCCESS
2. PROFIT
3. GROWTH
Customer investment

TYPICAL

20% Existing Customers
80% New Customers

DevFactory

20% New Customers
80% Existing Customers
Our Mission:

100% Customer Success
Are you being successful with FogBugz?
Great support helps drive customer success

1. Customer choice (3 support tiers)
2. World-class support talent (top 1% from across the globe)
3. Extended coverage (more reps, global coverage, and 24/7 support)
We’ve created support programs to help...

- **Standard**
  - 5 email + 1 phone per month

- **Gold**
  - 10 email + 5 phone per month
  - Quarterly Training for Dev Teams
  - Workflow Development Assistance

- **Platinum**
  - Unlimited email and phone support
  - 24x7 for Dev 1
  - 40 hours of Migration Import Assistance
  - Version Upgrade Service
  - On-prem to Cloud Migration Service
  - On-prem Update Assistance Test Environment
  - Older Version Support
  - Quarterly Training for Dev Teams
  - Workflow Development Assistance
# Support Details

<table>
<thead>
<tr>
<th></th>
<th>Standard</th>
<th>Gold</th>
<th>Platinum</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Customer Success</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Participation in Customer Success Program</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Quarterly Virtual User Group Conferences</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Product Development</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access to Patch and Update Client Product Releases</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Feature Request Priority Weighting</td>
<td>1x</td>
<td>2x</td>
<td>4x</td>
</tr>
<tr>
<td><strong>Customer Support</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phone Support during Business Hours</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Support Chat Capability during Business Hours</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Web-Based Reactive Ticketing + Phone (Month)</td>
<td>5 + 1</td>
<td>10 + 5</td>
<td>Unlimited</td>
</tr>
<tr>
<td>Web-Based Proactive Ticketing</td>
<td></td>
<td>Unlimited</td>
<td>Unlimited</td>
</tr>
<tr>
<td>Case Priority Weighting for Non-Critical Issues</td>
<td>1x</td>
<td>2x</td>
<td>4x</td>
</tr>
<tr>
<td>Support Availability (Hours x Days/Week)</td>
<td>6x8 EST</td>
<td>6x8 EST</td>
<td>24x7 for sev1</td>
</tr>
<tr>
<td>Version Support</td>
<td>8.20</td>
<td>8.19, 8.20</td>
<td>8.8, 8.19, 8.20</td>
</tr>
<tr>
<td>Platinum Experience Manager Reviews</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Product Experience</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quarterly Training for Dev Teams</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Workflow Development Assistance</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>40 hours of Migration Import Assistance</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Version Upgrade Service</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>On-prem to Cloud Migration Service</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>On-prem Update Assistance Test Environment</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
New Platinum Support

Included:
- 40 hours of Migration Import Assistance
- Version Upgrade Service
- On-prem to Cloud Migration Service
- On-prem Update Assistance Test Environment
- Older Version Support
- Quarterly Training for Dev Teams
- Workflow Development Assistance
Which of the support model offerings are you most interested in?
Access to pre-release DevFactory products

DevSpaces

CodeRead

CodeCleanup
The problem with dev environments today

- Container technology and Docker are the next big enablers for Software.
- Dockerization of apps is hard and everyone starts with their production environment
- Disconnect between the Dev and the production environments persist.
DevSpaces - What is it?

- It ‘shifts left’ and brings Docker to your dev environment
- It is a system that hosts your dockerized dev environments in the cloud without the need to configure the environment on their local workstations.
- It enables easy sharing of the standardized development environment across your team.
- An agent installed on your workstation syncs your local code instantly with the remote container - You can therefore use your favourite IDE to code locally, but build and run the app on the remote container.
- Our factory will dockerize your dev environment for you.
The problem with code reviews today

Christ people. This is just sh*t.

The conflict I get is due to stupid new gcc header file crap. But what makes me upset is that the crap is for completely bogus reasons. This is the old code in net/ipv6/ip6_output.c:

```c
mtu = hlen + sizeof(struct frag_hdr);
```

and this is the new `improved` code that uses fancy stuff that wants magical built-in compiler support and has silly wrapper functions for when it doesn’t exist:

```c
if (overflow_usub(mtu, hlen + sizeof(struct frag_hdr), &mtu) || mtu <= 7)
goto fail_too_big;
```

and anybody who thinks that the above is

(a) legible
(b) efficient (even with the magical compiler support)
(c) particularly safe

is just incompetent and out to lunch.

The above code is sh*t, and it generates sh*t code. It looks bad, and there’s no reason for it.

- Linus Torvalds
Code Reviews

- Fact 1: Code reviews are a judgment of code
- Fact 2: In most teams, the senior most perform code reviews
- Fact 3: Over a period of time, reviews degrade into 2 categories:
  - They only validate trivial formatting and static analysis rules
  - They become extremely judgemental and caustic

Code reviews are great, but you should never do them internally because at best they ruin the HR dynamics of your team
CodeRead - What is it?

- Don't do code reviews internally, we will do them for you - within 24 hours
- A dedicated team of senior architects review code
- AI assisted tool sets help identify the most important parts of the code that need attention
- Sentiment analysis gates and standardized review templates weed out judgement and turn reviews into learning/coaching experiences
The code catches any exception that may occur, then logs it and re-throws it.

An important improvement would be to use a custom exception filter to handle such exceptions. The change would avoid the computational cost of catching, handling, and re-throwing the exception, since exception filters do not unwind the stack.

More details

The solution in this case

Step 1: Create a (reservation) method so that it returns (false by default)

private boolean isReservationAvailable(String reservation, String status) {
    // create a new connection and a statement to check for availability
    try {
        Statement stmt = connection.createStatement();
        // execute the query
        ResultSet rs = stmt.executeQuery("SELECT status FROM reservations WHERE code = \\
                                        " + reservation);
        // check if the status is available
        if (rs.next() && rs.getString("status") != status) {
            // mark the status as available
            Database.updateStatus(reservation, status, "available");
            return true;
        } else {
            // mark the status as reserved
            Database.updateStatus(reservation, status, "reserved");
            return false;
        }
    } catch (SQLException e) {
        // handle database exceptions
        e.printStackTrace();
    } finally {
        // close the connection
        connection.close();
    }
    return false;
}

Step 2: Change the (reservation) block so that it uses that method in an exception filter:

catch (Exception e) {
    // use the custom exception filter
    handleReservationException(e);
}

NOTE:

- Consider applying the structure of this solution in other parts of the code. If there is another occurrence of

A good addition please implement as we can refer to it as best practice in the future.
The problem with technical debt today

- Project managers are constantly conflicted on allocating precious resources to fixing technical debt vs. working on features that add value to the product/customer.

- Technical Debt slows down the pace of development, so you can’t ignore it either.

- The top 3 contributors that slow down the effective pace of development are:
  - Duplicate Code
  - Long Methods
  - Memory Leaks

- None of these issues require domain or product knowledge to fix
CodeCleanup - The tech debt garbage collector

- Code Cleanup is a garbage collection service for technical debt in your code base.
- Use 10% of your R&D budget to clean up your code in the background.
- Enables your team to use their domain and product knowledge to add value to the product.
CodeCleanup on the FogBugz code today...

- **298 sources of Memory leaks** - impacts the stability and performance
- **191K lines of Duplicate code** - that implies that every time we fix an issue in one place, we have to be cognizant of also fixing the issue in all other instances of the duplication.
- **375 instances of Long methods** that average about 215 lines each - these methods contain important logic but are hard to understand & maintain because of their length and intertwined flows (cyclomatic complexity)

<table>
<thead>
<tr>
<th>Insight Type</th>
<th>#Non-Test Issues</th>
<th>#Test Issues</th>
<th>#Total Issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>Long Method</td>
<td>375 (80875 LOC)</td>
<td>21 (5217 LOC)</td>
<td>396 (86092 LOC)</td>
</tr>
<tr>
<td>Dead Code</td>
<td>8 (14 LOC)</td>
<td>0 (0 LOC)</td>
<td>8 (14 LOC)</td>
</tr>
<tr>
<td>Memory Leak</td>
<td>298 (298 LOC)</td>
<td>184 (184 LOC)</td>
<td>482 (482 LOC)</td>
</tr>
<tr>
<td>Long Class</td>
<td>554 (387075 LOC)</td>
<td>46 (26712 LOC)</td>
<td>600 (413787 LOC)</td>
</tr>
<tr>
<td>CodeRead Severe Insight</td>
<td>10080 (10080 LOC)</td>
<td>1063 (1063 LOC)</td>
<td>11143 (11143 LOC)</td>
</tr>
<tr>
<td>Duplicate Code</td>
<td>555 (191340 LOC)</td>
<td>32 (3021 LOC)</td>
<td>587 (194361 LOC)</td>
</tr>
</tbody>
</table>

Over the next few months, we will be working towards fixing these really important maintainability issues with the codebase.
Which of these DevFactory products are you most interested in?
Interested in early access? Get in touch!

Manuscript’s customers have exclusive early-bird access to DevSpaces, CodeRead, and CodeCleanup.
Agenda

About the Acquisition & what does it mean for me?

Why did we acquire Manuscript?

Future of FogBugz & DevFactory Products

Q&A and Summary
We bought FogBugz to have happy customers for the next 20 years...

This plan will create the foundation for the next 10 years of innovation.