Replace TPL with Python and IDE

Suggestion

***Gradually** replace TPL with Python and a proper Python IDE.*

There is a huge opportunity to improve the capability of BMC Discovery for Pattern Devepment. Most of the suggestions (ideas) on this site are fairly conservative incremental ideas that do not properly address the probllem.   A more radical approach is to replace TPL with Python and a proper Python IDE. ***Please don't immediatiely dismiss this idea as crazy***. I know it is radical - but in the long run - this approach could well be less expensive than other options and will lead the product in hte right direction. Please read my thoughts below.

Other Similar Ideas on the Communities

There are many similar ideas on the Communities that address the same area. This indicates that this topic is a common concern for many Users.
Most of the ideas suggested in the Ideas below are fairly conservative in nature (such as using Notepad++) and only address one of the requirements (see table below).

TPL Syntax Highlighting for Visual Studio Code
IDE for Patterns (and Other Ideas)
IDE for TPL
Experimental development IDE for TPL
TPL Syntax Highlighters
TPL Code Hints and Syntax Highlighting (for Brackets)
TPL development framework
TPL Validation Method?
TPL syntax "helper"
Pattern/TKU Deployment via API - using continuous integration/automation tools like Jenkins?

Key Requirements

Number	Requirement	Description
1	Fail Fast	In terms of productivity, the biggst factor is re-execution time between code edits. Currently,

		thee re-execution time is 1-2 minutes; it should be 10 seconds.
2	IDE	An IDE with features for refactoring, automatic syntax highlighting, coding standards check,
4	Unit Testing	Essential for modern TDD.
5	GIT Integration	Would promote a thriving Pattern Development Community.
6	Modules and Classes	A Proper objected orientated language is essential for modern progamming.
7	pydoc	Automatic documentation would be cool. It could be viewable in the Console.
8	ADDM Class Library	With a python IDE on the Appliance, Users could access an ADDM Class Library (which would need to be published and documented).

Migrate to Python and PyCharm

The time and resources to develop a proper TPL Development environment might be more than migrating to Python - and then allowing users to install pyCharm (or something similar) on the Appliance. If you try and develop IDE type features for TPL - they will always tend to be sub-standard compared to pyCharm. The benefit of using Python would be:

Benefits of using Python

- Proper Unit Testing
- Access to a great IDE such as PyCharm.
- Acces to the full Appliance class library (BMC would need to publish and document these)
- Integration to GIT (via PyCharm)
- The possibility that a thriving Customer Development community would evolve and help with Pattern Development.
- The cost of writing and support a custom IDE for TPL would be huge (and a cut down tool would always be subpar). If TPL is parsed into Python already and the Python class linbray already exists, then this option might not be that expensive. I am guessing here?
- If the capability could be provided in parallel, then migrating of existing patterns could be done over a few years.

- The possbility of Innovative Customer Plugins - that could become part of the product.
- Grater product quality and market share for ADDM.
- Greater profit for BMC.
- Happiness and world peace.

Objection Handling

The key objections will are the following:

Number	Objection	Handling
1	Its would be too hard to develop.	I am am not privy to the design and architecture of BMC discovery - so I have no answer to this objection. I have always assumed that most of the code in ADDM is developed in Python and TPL is simply parsed into Python and then executed by the python engine. I am hoping that this is correct. If this is correct, then the main task would be to sort out and document a proper python class library that could be made public. This does not sound too hard - but I am making a lot of assmptions which could be wrong.
2	Open Source Code would be stolen	Yes. Comptetitors could "acquire" commercialy sensitive patterns (if they were written in Python). It would be interesting to understand the orginal reason why TPL was chosen. Why not Python? I guess that the attitude to Open Source code was differnt 20 years ago. Today, many Software companies have made their code "Open Source" and it has worked. I think the benefits of Open Source would outweight the disadvantages.
4	Market Share? Cash Cow or Star?	From a benefits point of view, I think it is a no brainer. There real decision is "Is it possible and cost justifiable?" I guess from a Boston Consulting Group Portfolio model point of view, is BMC a Cash Cow, a Dog or a Star? I like the product. I give it 3-4 stars. It could be 5 - but there needs to be a commitment. Is the

		market for BMC Discovery only Remedy Customers? (btw: we are a ServiceNow Customer). Is this market, rising or falling? Can BMC (realistically) increase market share that much? I think it can - but ADDM must be a 5 star product. These are all interesting Questions and I am not provy to a lot of facts - so I don't have many answers. At the end of the day, someone in change has got to trust their instincts and make a decision.