FIA | Principal Traders Group
European Principal Traders Association

Software Development and Change Management Recommendations

Introduction

This document was created by a joint working group of the Futures Industry Association ("FIA") Principal Traders Group and the FIA European Principal Traders Association (collectively "FIA PTG & EPTA"). FIA PTG & EPTA are forums for firms trading their own capital to identify and discuss issues confronting the principal traders' community. Membership in both groups is limited to firms that trade exclusively for their own account. The groups work to define common positions on public policy issues and advance the groups' collective interests through the FIA; improve public understanding of the constructive role played by principal trading firms in the exchange-traded derivatives markets; and promote cost-effective, equal and transparent access to markets. FIA PTG & EPTA members engage in manual, automated and hybrid methods of trading on exchanges located around the world and are active in cash and derivatives in a variety of asset classes, such as equities, foreign exchange, commodities and fixed income.

Software is at the core of many business critical activities within FIA PTG & EPTA member firms. For a firm to meet its regulatory obligations and manage and monitor risks, its trading software and technical infrastructure need to work as intended. To support trading firms in meeting this objective, the joint working group has developed best practices for software development, testing, and change management. These best practices will assist trading firms in developing their own procedures related to software development.

This document does not attempt to recommend specific policies and procedures for firms. Specific procedures will vary depending on the firm's size, scope, and trading strategies, and the requirements associated with the markets in which the firm operates.

There is a large body of work that addresses the topic of software best practices and we do not attempt to recreate or reiterate that body of work here. Instead, our purpose is to offer principles to guide trading firms in establishing their internal procedures, processes, and controls for the development, testing, and deployment of trading software. In applying these principles, source code implementing the following functionality may merit additional attention:

- Trading infrastructure

- Risk controls

- Compliance functionality

Properly established procedures and lines of communication assist firms in mitigating the risk of errors associated with the introduction of software changes. Such procedures should always be designed to meet the unique needs of an individual firm.

The best practices provided herein should be considered in conjunction with *FIA Principal Traders Group: Recommendations for Risk Controls for Trading Firms* (*FIA PTG Recommended Risk Controls*).[1] Taken together, these documents provide a framework for risk mitigating practices that span the entire software lifecycle.

---

[1] http://www.futuresindustry.org/downloads/Trading_Best_Pratices.pdf

Although the primary audience for this document is firms that directly access exchange trading environments, many of the best practices are broadly applicable to the entire trading community, which today is heavily reliant on software. Finally, although the term "best practices" is used in this paper, it should be noted that software best practices have and continue to evolve. It is, therefore, important that trading firms periodically evaluate their policies and practices and update them as appropriate.

Software Development

Trading firms should have a process in place through which they can implement new code or changes to existing source code. A variety of software development methodologies exist and each firm should employ a methodology that promotes efficient communication, generates maintainable source code, and produces software that is implemented to specification. Best practices for software development processes should address the following:

Development Environment—Firms should maintain a development environment that is adequately isolated from the production trading environment. The development environment may include computers, networks and databases and should be used by software engineers while developing and testing new source code.

Source Code Management—Firms should maintain a source code repository to manage source code access, persistence, and changes. The source code repository may be used to ascertain when software changes were made and the nature of the changes.

Risk Controls—Firms should implement software based risk controls that are independent from the trader in order to reduce the risk of market disruptions due to system failures or errors (see *FIA PTG Recommended Risk Controls*).

Source Code Review—When appropriate, firms should have a process describing how software engineers may have their source code reviewed and how that review may be conducted.

Software Testing

Trading firms should have a process for testing core software components before they are released to the production environment. Software testing should be appropriate and proportionate to the change being made. A variety of effective testing methodologies exist and each firm should employ a suite of software testing tools to suit their unique needs. Among the testing methods to consider are:

Unit Testing—A type of testing in which discrete units of source code are tested to verify they work as desired. These tests may be configured to run automatically throughout the development process.

Functional Testing—A type of testing in which well-defined software modules are combined to have their functionality tested as a group. Two types of functional testing that may be considered are "integration" and "regression" testing.

Non-Functional Testing—A type of testing in which well-defined software modules are combined to have their non-functional aspects tested as a group. Such non-functional aspects might include scalability, performance, stability, and usability.

Acceptance Testing—A type of testing in which the software is tested by an end-user to verify conformance of a system to the stated business requirements. Acceptance testing should be done in an environment that adequately represents the environment in which the software will be released.

Exchange-Based Conformance Testing—A type of testing utilized to confirm a system's functionality while interacting with an exchange. This process is often guided by a script of tests provided by the exchange and is performed in an exchange-provided testing environment to simulate the production trading environment.

During the testing process, firms should consider potential impact to trading systems, external markets, compliance systems, middle and back office systems, user interfaces, and reporting mechanisms.

Change Management Process

Trading firms should have an established change management process that is designed to control and manage the propagation of software and infrastructure changes to a production environment.

The following principles can serve as building blocks firms may use to tailor a change management process to best fit their needs. Firms may choose to combine one or more of the steps into a single step in their process, while others might elect to split a particular step into several other sub-steps. Different kinds of changes may warrant different refinements or variants of the process, depending on the nature of the changes and their potential impacts.

Change Management Core Components

The following practices are integral to a trading firm's change management process:

- **Authorization**—Any changes to the production environment should be subject to review by a responsible party within the organization. The depth of the review performed should align with the magnitude of the proposed change.

- **Auditability**—Trading firms should establish procedures for communicating requirements, changes and functionality related to their proprietary software and technical infrastructure. Trading firms should also maintain a historical audit trail of material changes made to their proprietary software, allowing them to accurately determine:

  - When a change was made

  - Who made the change

  - The nature of the change

Steps Commonly Seen Within the Release Process

- **Initiation**—Every software change is initiated to meet a business, technical, or external requirement. The initiator of the change should identify the requirement(s) or nature of the change.

- **Approval**—Prior to deployment, a planned change should be reviewed and subject to approval by a responsible party. This review may occur prior to development taking place or after development is completed.

- **Scheduling**—Prior to deployment, a planned change should be scheduled for release into the production environment, and should be considered along with any other planned changes.

- **Deployment**—Deployment is the act of releasing a change into the production environment. Depending on the nature of the change, it may be appropriate to deploy to the entire production environment at once or to deploy the change in phases to further mitigate risk and ease the reversion of the change if necessary.

  Deployment may be thought of as containing four phases:

  (1) **Preparation**—The change is prepared for release and the current production environment is backed up in order to allow for reversion of the change.

  (2) **Execution**—The change is released to the production environment.

  (3) **Validation**—The change and the state of the production environment should be verified for correctness. The scope of a firm's validation process should be appropriate and proportionate to the change being made.

  (4) **Completion/Reversion**—A successful validation should result in completion of the change. If the change cannot be validated, the environment should be reverted to its prior stable state.

- **Post Deployment**—Special consideration should be given to how certain changes to trading systems may impact trading in the production environment. Where reasonable, substantive changes to trading systems should be activated initially with appropriately restricted risk limits and access to markets.

These practices facilitate effective risk management and are consistent with the overall development and change management process: identifying the desired or required change, developing and testing the change, deploying the change, and verifying the change.

Security

Trading firms should establish security measures within all aspects of their business. In addition to the security measures in the *FIA PTG Recommended Risk Controls*, some considerations in building effective security measures for software development and change management include:

- Maintaining source code, technical infrastructure and trading systems ("Technology") in a physically, technologically and otherwise secure manner; and

- Allowing access to Technology to approved persons and through mechanisms that validate identity in a manner consistent with a firm's regulatory obligations and internal requirements.

Conclusion

The best practices outlined in this paper will assist trading firms in managing the complex and dynamic process of software and infrastructure development, testing, and deployment. Proper application of these best practices will reinforce an FIA PTG & EPTA member firm's role as a responsible market participant who is committed to business practices that are consistent with relevant regulatory obligations and risk management requirements.

Ongoing dialogue between FIA PTG & EPTA member firms will continue evolving these best practices as dictated by changes in technology and the needs of the marketplace. The FIA PTG & EPTA welcome feedback from other industry participants, including relevant regulatory bodies, exchanges, technology providers, and other market stakeholders.